

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Robert Pavlović

# **Tečaj postavitve oblačne infrastrukture**

DIPLOMSKO DELO  
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: dr. Andrej Brodnik

Ljubljana 2014



Rezultat diplomskega dela so intelektualna lastnina avtorja. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakulteta za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Vedno večim uporabnikom predstavlja oblačna infrastruktura okolje, v katerem se nahajajo najrazličnejše storitve, ki jih uporabljajo. Običajno je to javna infrastruktura ali javni oblak. Hkrati postaja postavitve zasebne infrastrukture ali zasebnega oblaka tudi zanimiva za nekatere uporabnike. V ta namen obstajajo tako zaprtokodne, kot odprtokodne in predvsem brezplačne rešitve. Vendar postavitve infrastrukture zahteva nekatera posebna znanja in spretnosti, ki so sicer na voljo na svetovnem spletu a pogosto v precej razpršeni obliki.

V diplomski nalogi preučite splošne lastnosti oblačne infrastrukture in lastnosti njenih najpomembnejših gradnikov kot je virtualizacija. Nato pripravite začetniški e-tečaj postavitve zasebnega oblaga na minimalni infrastrukturi ter ovrednotite kakovost e-tečaja.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Robert Pavlović, z vpisno številko **63080196**, sem avtor diplomskega dela z naslovom:

*Tečaj postavitve oblačne infrastrukture*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom dr. Andreja Brodnika,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 16. septembra 2014

Podpis avtorja:





*Zahvaljujem se mentorju dr. Andreju Brodniku za strokovno pomoč, kvaliteten odnos, predloge in izboljšave. Zahvaljujem se tudi osebju Laboratorija za vseprisotne sisteme za pomoč in usmeritve ter osebju Laboratorija za računalniške komunikacije za postavljeno infrastrukturo in pomoč. Posebna zahvala gre moji družini, ki me je zelo podpirala ter mi stala ob strani tekom študija. Zahvaljujem se tudi vsem ostalim, ki so pripomogli k boljšemu rezultatu diplomske naloge.*



*Diplomsko delo posvečam svojim staršem.*



# Kazalo

**Povzetek**

**Abstract**

<b>1</b>	<b>1 Uvod</b>	<b>1</b>
1.1	Uvod v diplomsko delo . . . . .	1
1.2	Motivacija . . . . .	3
1.3	Struktura diplomskega dela . . . . .	5
<b>2</b>	<b>Učenje v e-izobraževanju</b>	<b>7</b>
2.1	Psihološke študije in uspešnost učenja . . . . .	7
2.2	Klasičen pristop in e-izobraževanje . . . . .	10
2.3	Sistemi za upravljanje e-izobraževanj . . . . .	12
2.4	Priprava tečaja . . . . .	13
<b>3</b>	<b>Računalništvo v oblaku</b>	<b>25</b>
3.1	Razvoj računalništva v oblaku . . . . .	25
3.2	Opredelitev računalništva v oblaku . . . . .	29
3.3	Storitveni modeli računalništva v oblaku . . . . .	31
3.4	Nadzorni modeli računalništva v oblaku . . . . .	34
3.5	Splošna arhitektura infrastrukture . . . . .	36
3.6	Prednosti in nevarnosti uporabe oblaka . . . . .	38
<b>4</b>	<b>Virtualizacija kot sestavni del infrastrukture oblaka</b>	<b>41</b>

## KAZALO

4.1	Resnični stroj . . . . .	41
4.2	Navidezni stroj . . . . .	43
4.3	Razvoj virtualizacije . . . . .	45
4.4	Hipernadzornik . . . . .	46
4.5	Virtualizacija in virtualizacijske tehnike . . . . .	47
4.6	Prednosti in slabosti virtualizacije . . . . .	58
<b>5</b>	<b>Okolje OpenStack</b>	<b>61</b>
5.1	Predstavitev OpenStacka . . . . .	61
5.2	Komponente OpenStacka . . . . .	63
5.2.1	Računska storitev Nova . . . . .	63
5.2.2	Identifikacijska storitev Keystone . . . . .	63
5.2.3	Storitev hrambe slik diskov Glance . . . . .	64
5.2.4	Omrežna storitev Neutron . . . . .	64
5.2.5	Storitev za shranjevanje blokovnih naprav Cinder . . . . .	66
5.2.6	Storitev za shranjevanje objektov Swift . . . . .	67
5.2.7	Spletni vmesnik Horizon . . . . .	67
5.2.8	Storitev za izvajanje opazovanja Ceilometer . . . . .	68
5.2.9	Storitev orkestracije Heat . . . . .	68
5.2.10	Podatkovna baza kot storitev Trove . . . . .	69
5.3	Logična arhitektura platforme OpenStack . . . . .	69
<b>6</b>	<b>Tečaj postavljanja okolja OpenStack</b>	<b>71</b>
6.1	Uvod v tečaj . . . . .	71
6.2	Naloga 1: Preverjanje sistemskih zahtev OpenStacka in uvod v Ubuntu Linux . . . . .	73
6.3	Naloga 2: Namestitve in nastavitve komponent OpenStack . . . . .	74
6.4	Naloga 3: Vzpostavitev okolja OpenStack . . . . .	75
6.5	Naloga 4: Zagon navideznega stroja . . . . .	76
<b>7</b>	<b>Evalvacija tečaja</b>	<b>77</b>
7.1	Cilj raziskave . . . . .	77

## KAZALO

7.2	Metoda raziskovanja in raziskovalni vzorec . . . . .	78
7.3	Postopek zbiranja in obdelave podatkov . . . . .	81
7.4	Interpretacija rezultatov . . . . .	82
7.5	Preverjanje zastavljenih učnih ciljev . . . . .	87
8	<b>Zaključek in nadaljnje delo</b>	<b>91</b>
	<b>Dodatki</b>	<b>95</b>
A	<b>Zgoščenska</b>	<b>97</b>
B	<b>Primer navidezne infrastrukture</b>	<b>99</b>
C	<b>Konceptualna arhitektura platforme OpenStack</b>	<b>101</b>
D	<b>Procesi komponent</b>	<b>103</b>
E	<b>Logična arhitektura platforme OpenStack</b>	<b>113</b>
F	<b>Vprašalnika</b>	<b>115</b>

*KAZALO*



# Slike

2.1	Interakcija pri klasičnem učenju med učiteljem in učencem. . .	10
2.2	Interakcija med učencem in učnim gradivom v LMS-ju. . . . .	11
2.3	Teoretična učna enota <i>Uvod v računalništvo v oblaku</i> . . . . .	16
2.4	Praktična učna enota <i>Preverjanje sistemskih zahtev OpenStacka in uvod v Linux Ubuntu</i> . . . . .	17
3.1	Razvoj računalništva v oblaku skozi čas. . . . .	29
3.2	Osnovni storitveni modeli oblaka. . . . .	33
3.3	Postavitveni modeli računalništva v oblaku. . . . .	36
3.4	Splošna arhitektura storitvenega modela IaaS. . . . .	38
4.1	Arhitektura von Neumannovega računalnika. . . . .	43
4.2	Navidezni računalnik za svoje delovanje uporablja navidezne vire. . . . .	44
4.3	Umestitev hipernadzornika in njegovo delovanje. . . . .	47
4.4	Aplikacijska virtualizacija programskega jezika Java. . . . .	49
4.5	Virtualizacija omrežja. . . . .	51
4.6	Virtualizacija hrambe. . . . .	52
4.7	Emulacija strojne opreme kot strežniška virtualizacija. . . . .	53
4.8	Polna virtualizacija kot strežniška virtualizacija. . . . .	54
4.9	Paravirtualizacija kot strežniška virtualizacija. . . . .	55
4.10	Virtualizacija na nivoju operacijskega sistema. . . . .	56
4.11	Strojno podprta virtualizacija. . . . .	58

5.1	Različna omrežja znotraj oblaka OpenStack. . . . .	66
5.2	Spletni vmesnik za upravljanje OpenStacka. . . . .	68
7.1	Značilnost obeh vzorcev glede na oceno predznanj in izkušenj. . . . .	81
7.2	Ocene stopenj strinjanj s trditvami o izvedbi e-tečaja. . . . .	83
7.3	Ocene stopenj strinjanj s trditvami o praktičnem delu tečaja. . . . .	85
7.4	Rezultati pravilno opravljenih nalog. . . . .	87

# Tabele

7.1	Značilnost vzorca desetih učencev. . . . .	79
-----	--	----



# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>AaaS</b>	Application as a Service	aplikacija kot storitev
<b>AMD-V</b>	AMD Virtualization Technology	AMD-jev tehnologija za podporo stroje virtualizacije
<b>AMQP</b>	Advanced Message Queuing Protocol	napredni komunikacijski protokol
<b>API</b>	Application programming interface	aplikacijski programski vmesnik
<b>AWS</b>	Amazon Web Services	Amazonove storitve v oblaku
<b>CaaS</b>	Computing as a Service	računanje kot storitev
<b>CLI</b>	Command Line Interface	vmesnik z ukazno vrstico
<b>CPU</b>	Central Processing Unit	centralna procesna enota (CPE)
<b>DHCP</b>	Dynamic Host Configuration Protocol	protokol za dinamično dodeljevanje naslovov IP
<b>EC2</b>	Amazon Elastic Compute Cloud	Amazonov oblak za ponujanje infrastrukture kot storitve
<b>EMC</b>	Egan Marion Company	Ameriško podjetje
<b>GB</b>	Gigabyte	enota informacije, velikost podatkov

<b>kratica</b>	<b>angleško</b>	<b>slovensko</b>
<b>GUI</b>	Graphical User Interface	grafični uporabniški vmesnik
<b>HDD</b>	Hard disk drive	diskovni pogon
<b>HP</b>	Hewlett-Packard	Ameriško podjetje
<b>HTTP</b>	HyperText Transfer Protocol	protkol za prenos nadbese- dila na svetovnem spletu
<b>IaaS</b>	Infrastructure as a Service	infrastruktura kot storitev
<b>IBM</b>	International Business Ma- chines	Ameriško podjetje
<b>ID</b>	Identifier	označevalnik
<b>INFORMS</b>	The Insitute for Operations Research and the Manage- ment Science	mednarodno združenje za operativne raziskave in upra- vljanje znanosti
<b>IP</b>	Internet Protocol	internetni protokol za komu- nikacijo
<b>IT</b>	Information Technology	informacijska tehnologija
<b>KVM</b>	Kernel-based Virtual Ma- chine	izvajanje virtualizacije na osnovi jedra OS-a
<b>L3</b>	Layer 3	programska oprema za omrežje
<b>LMS</b>	Learning management sy- stem	sistem za upravljanje izo- braževanja
<b>MB</b>	Megabyte	enota informacije, velikost podatkov
<b>Moodle</b>	Modular Object-Oriented Dynamic Learning Enviro- nment	izobraževalno e-okolje v obliki spletne aplikacije
<b>NASA</b>	Network Address Transla- tion	Nacionalna zrakoplovna in vesoljska uprava
<b>NAT</b>	Network address translation	spreminjanje omrežnega na- slova IP

<b>kratica</b>	<b>angleško</b>	<b>slovensko</b>
<b>NIC</b>	Network interface controller	mrežna kartica
<b>OS</b>	Operating system	operacijski sistem
<b>PaaS</b>	Platform as a Service	platforma kot storitev
<b>PHP</b>	Hypertext Preprocessor	skriptni programski jezik, ki se tolmači na strani strežnika
<b>PNI</b>	Physical Networking Infra- structure	fizična infrastruktura omrežja
<b>QEMU</b>	Quick EMUlator	odprtokoni program za emulacijo virtualizacije računalnikov
<b>RAID</b>	Redundant array of indepen- dent disks	redundantno diskovno polje
<b>RAM</b>	Random-Access Memory	bralno-pisalni pomnilnik
<b>REST API</b>	Representational State Transfer API	programski vmesnik zasno- van na arhitekturnem slogu REST
<b>RPC</b>	Remote Procedure call	klic za oddaljeni postopek
<b>SaaS</b>	Software as a Service	programska oprema kot sto- ritev
<b>SOAP</b>	Simple Object Access Proto- col	protokol za spletne storitve
<b>SQL</b>	Structured Query Language	strukturirani povpraševali je- zik
<b>SSH</b>	Secure Shell	varnostna lupina
<b>STaaS</b>	Storage as a Service	hramba kot storitev
<b>S3</b>	Simple Storage Service	Amazonova storitev za hrambo podatkov v oblaku
<b>vHDD</b>	Vitrual HDD	navidezni diskovni pogon
<b>VLAN</b>	Virtual LAN	navidezno lokalno omrežje

kratica	angleško	slovensko
<b>VLE</b>	Virtual learning environment	navidezno učno okolje
<b>VNC</b>	Virtual Network Computing	protkol za oddaljeni dostop
<b>VNI</b>	Virtual Networking Infrastructure	navidezna infrastruktura omrežja
<b>vNIC</b>	Virtual NIC	navidezna omrežna kartica
<b>VPN</b>	Virtual private network	navidezno zasebno omrežje
<b>vRAM</b>	Virtual RAM	navidezni bralno-pisalni pomnilnik
<b>VT-x</b>	Virtualization Technology	Intelova tehnologija za podporo strojne virtualizacije
<b>WSDL</b>	Web Service Description Language	opisovalni jezik spletnih storitev
<b>WSGI</b>	Web Server Gateway Interface	vmesnik med spletnim strežnikom in aplikacijami napisanimi v programskem jeziku Python



# Povzetek

Glede na aktualnost uporabe računalništva v oblaku, še posebej infrastrukture kot storitve, smo v okviru diplomskega dela izdelali e-tečaj za vodeno izobraževanje gradnje oblakov na osnovi platforme OpenStack, ki neizkušenim skrbnikom sistemov omogoča hiter in enostaven uvod v računalništvo v oblaku ter poznavanje osnovne namestitve platforme. Tečaj omogoča tudi številne možnosti nadgrajevanja znanja v obliki samostojnega učenja. V prvem delu smo s pomočjo psihološkega pristopa k učenju in smernic za izdelavo e-izobraževanja zastavili oblikovno ter vsebinsko zgradbo tečaja. Tečaj smo razdelili na teoretične in praktične učne enote. V sklopu teoretičnih enot smo predstavili značilnosti, storitvene in postavitvene modele računalništva v oblaku, nadaljevali smo z virtualizacijo, ki je bistveni del infrastrukture oblaka, nato smo se seznanili z delovanjem platforme. V sklopu praktičnega dela tečaja, ki smo ga zaradi vodene učne poti in lažjega razumevanja razdelili na štiri naloge, smo opravili osnovno namestitev in nastavitve platforme. Po izvedbi tečaja smo v empiričnem delu naloge izdelali evalvacijo tečaja. Opravili smo raziskavo na manjši skupini učencev, ki so podali sistematično oceno njegove vrednosti na podlagi preverjanja doseženih učnih ciljev, pravilne vsebinske in oblikovne zasnove tečaja ter nekaj pomanjkljivosti, ki se v prihodnosti lahko uporabijo kot možnosti izboljšav. Ocena tečaja je na splošno dobra. Veliko boljša bi bila, če bi pri smernicah namesto priporočil izdelave e-tečaja opravili manjšo raziskavo o predznanju učencev. Vse aktivnosti tečaja so se izvajale v spletni učilnici Moodle.

**Ključne besede:** računalništvo v oblaku, infrastruktura kot storitev, e-izobraževanje, vodeni tečaj, platforma OpenStack, zasebni oblak

.

# Abstract

In view of the actuality use of cloud computing, particularly infrastructure as a service, we have in the context of diploma thesis developed e-course for a guided education for building clouds based on OpenStack platform, which allows novice system administrators a quick and easy introduction to cloud computing and knowledge of the basic installation of platform. The course also provides a number of options for advancing knowledge for learners in the form of self-directed learning. In the first part we set design and content structure of the e-course on basics of a psychological approach of learning and guidelines for making e-learning. The course was divided into theoretical and practical learning units. Within the theoretical units were presented features, service and models of installation of cloud computing, we continued with virtualization, which is an essential part of the infrastructure of a cloud, then we are acquainted with the functioning of the platform. Within the practical part of the course, which was due to guided learning paths and easier understanding divided into four tasks, we performed a basic installation and setup of platform. After the execution of the course, we have made empirical part of the thesis which elaborated evaluation of the course. We have performed a study on a small group of students who have made a systematic assessment of its value on the basis of the examination achieving learning goals, proper design and concept of course and a few weaknesses which can be used as an opportunity for improvement in the future. In general the score of course is good, but it would be much higher if instead of using recommendations for designing e-course we would

use previous knowledge of learners. All activities of the course were performed in Moodle virtual classroom.

**Keywords:** cloud computing, infrastructure as a service, e-learning, guided course, OpenStack platform, private cloud.

# Poglavje 1

## 1 Uvod

### 1.1 Uvod v diplomsko delo

Zahvaljujoč nenehnemu napredku informacijske tehnologije na področju računalništva v oblaku je moč zaslediti naraščajočo prisotnost oblakov tako s strani ponudnikov storitev, kot tudi podjetij in navadnih uporabnikov, ki te storitve uporabljajo. Za uporabnike storitev je najpomembneje, da so storitve vedno na voljo, da do njih lahko dostopajo ne glede na lokacijo in čas ter da delujejo brez napak. Pri tem potrebujejo le hitre in varne internetne povezave. Podjetja, ki storitve ponujajo, morajo zagotoviti omejene zahteve uporabnikov, saj je njihovo poslovanje popolnoma odvisno od informacijsko-komunikacijskih sistemov in v veliki meri od njihovih skrbnikov. Skrbniki sistemov so namreč nemalokrat preobremenjeni s postavljanjem in vzdrževanjem informacijskih sistemov, ki vključuje: nameščanje strežnikov, nadgrajevanje sistema, nameščanje in posodabljanje programske opreme ter upravljanje omrežja. Prav tako se računalniško usmerjena podjetja ukvarjajo z istim problemom, zato se odločijo, da svoj informacijski sistem v celoti ali pa deloma nadomestijo s storitvami, ki jih ponuja računalništvo v oblaku in se pri tem osredotočijo le na iskanje primernih storitev svojim potrebam, ceno in način obračunavanja storitev.

V kolikor na tržišču ni primerne storitve oblaka oziroma se podjetja ukvarjajo z dejavnostmi, kot so analiza in obdelava občutljivih podatkov in pri tem ne zaupajo ponudnikom oblaka, lahko za lastne potrebe ob obstoječi infrastrukturi postavijo zasebni oblak. V tem primeru imajo največ opravil na začetku, saj je oblak potrebno namestiti, testirati in nanj emigrirati podatke ter aplikacije. Nadaljnje delo pa je precej enostavnejše in hitrejše. Ravno na začetku uvajanja oblaka v svoje delovno okolje se pojavi največ težav. Izkušeni skrbniki in arhitekti sistemov s tem postopkom nimajo veliko težav, neizkušeni pa jih zagotovo imajo, saj pri slabem predhodnem poznavanju tako posameznih delov kot celotnega informacijskega sistema zgolj dokumentacija ne zadostuje.

Cilj diplomskega dela je izdelati tečaj za postavitev oblaka in omenjeni skupini v čim krajšem času predstaviti problematiko, jo seznaniti s teoretično platjo računalništva v oblaku ter ji omogočiti, da pridobljeno teoretično znanje skozi postopek namestitve in nastavitve platforme OpenStack [58] nadgradi. Tečaj je razdeljen na teoretični in praktični del. Na njuni podlagi učenci pridobijo ustrezna znanja za razumevanje konceptov računalništva v oblaku in so usposobljeni za samostojno osnovno postavitev ter uporabo oblaka. Pridobljeno znanje lahko nadgradijo tudi z naprednejšimi namestitvami in z nastavitvami pri gradnji oblaka, ki jih tečaj ponuja v obliki iztočnic s samostojnim učenjem. Vsaka praktična naloga opisuje predhodna potrebna znanja, učne cilje oziroma pridobljena znanja, navodilo naloge, praktični postopek z ukazi in s komentarji za boljše razumevanje ter možnosti nadgrajevanja znanja s samostojnim učenjem. Za izvedbo naloge je potrebno osnovno poznavanje računalnika in arhitekture računalniških sistemov. Platformo bomo namestili na operacijski sistem (v nadaljevanju OS) Ubuntu Linux [76]. Celoten tečaj se izvaja v spletni učilnici Moodle [47]. Diplomsko delo je empirično zasnovano, saj v zadnjem delu vključuje evalvacijo tečaja na manjši skupini učencev.

## 1.2 Motivacija

Funkcionalnost računalništva v oblaku je podobna porazdeljenemu računalništvu iz šestdesetih let. Po letu 2000 se je model računalništva v oblaku začel uveljavljati kot samostojna rešitev v informacijsko-komunikacijskih sistemih, saj predstavlja pomembnejšo tehnološko in poslovno prioriteto v podjetjih, ki so prepoznala številne prednosti njegove uporabe. Le-te so: večja osredotočenost na primarno dejavnost, hitrejša prilagodljivost poslovnih procesov, konkretno zniževanje operativnih stroškov informacijskih storitev, celovitih storitev in po meri ter način njihovega zaračunavanja. V računalniških podjetjih, katerih primarne dejavnosti so obdelava in analiza podatkov ter ponujanje storitev, se pojavlja velika negotovost glede varnosti in zasebnosti podatkov. Zato se veliko podjetij odloči za postavitev lastnega oblaka s pomočjo plačljivih ali odprtokodnih platform. Uvajanje oblaka v podjetja zahteva celovit pristop in v primeru neustrezno izobraženega lastnega kadra to ni enostavno opravilo. Večina prosto dostopnega gradiva je za skrbnike sistema z malo delovnimi izkušnjami in s slabo teoretično podlago neprimerno zastavljena, saj zahteva veliko predznanja, gradiva so preobsežna in veliko časa bi morali nameniti branju ter iskanju dodatnih informacij. Ta problem je mogoče rešiti na več načinov. Prva možnost je ustrezno usposobljen zunanji izvajalec, ki ponuja celovit pristop; druga možnost je izobraževanje lastnega kadra s plačljivimi delavnicami in tretja možnost je uporaba izdelanega tečaja.

V diplomskem delu Blaža Brenčiča, ki jo je opravil na Ekonomski Fakulteti Univerze v Ljubljani leta 2010 z naslovom » *Računalništvo v oblaku: Stanje v Sloveniji in primerjava s tujino*«, je raziskal prisotnost računalništva v oblaku v Sloveniji [15]. Na medmrežju nisem zasledil novejših tovrstnih raziskav, zato sem se odločil, da bom rezultate omenjene naloge uporabil v okviru svojih potreb. V raziskavo je vključil preko 200 slovenskih podjetij, z različno pravno obliko in različnimi dejavnostmi. V prvem delu je moč zaslediti, da 33,3 % podjetij ne načrtuje uporabe računalništva v oblaku, 21,4 % jih v

prihodnosti načrtuje, 21,4 % jih že uporablja storitve, 14,2 % so ponudniki teh storitev, preostalih 10,7 % bodo uporabili drugačne pristope (del raziskave vključuje 194 podjetij). Drugi del raziskave podaja način namestitve oblaka – zasebni oblak bi postavilo 33,5 % podjetij; 22,5 % bi jih uporabilo oblak, ki ga lasti partnersko podjetje; odstotek podjetij, ki bi uvedli hibridni oblak, je enako podjetjem, ki niso prepričani v to, kakšen nadzorni model bi uvedli v svoje okolje, in znaša 17,3 %; javni oblak bi uporabilo le 7,8 % podjetij, preostalih 1,6 % ni odgovorilo (del raziskave vključuje 191 podjetij). Glede na storitvene modele bi 34,9 % podjetij uporabilo infrastrukturo (IaaS), 28,9 % podjetij programsko opremo (SaaS), 22,6 % podjetij platformo (PaaS), ostalih 14,6 % podjetij bi uporabilo varnostne in druge storitve (del raziskave vključuje 175 podjetij). V Sloveniji je 69 podjetij (zdaj verjetno še več), ki že uporablja storitve oblaka. Najbolj jim ustreza Googlov nabor (37,7 %), le-tem sledijo domače rešitve (19,28 %), ostala globalna podjetja so zastopana v bistveno manjših deležih in niso vredna omembe.

V zadnjem času sta od odprtokodnih platform najbolj izpostavljena CloudStack [7] in OpenStack. Slednji je modularen, saj omogoča povečljivost, je neodvisen, komponente platforme se lahko izvajajo posamično ali pa skupaj, kar pripomore k visoki prožnosti sistema. Zaradi omenjenih prednosti je njegova velika težava nameščanje in vzdrževanje. CloudStack omogoča osnove funkcionalnosti platforme, bistveno lažje ga je namestiti, vendar ne ponuja povečljivosti na infrastrukturi. Oba podpirata velik nabor hipernadzornikov [56].

Rezultati raziskave nakazujejo, da slovenska podjetja ne zaupajo tujim ponudnikom storitev v oblaku, torej hočejo vzpostaviti zasebno okolje. Vedno več je tudi podjetij, ki bi uporabili infrastrukturo. Prej smo omenili, da je velika slabost OpenStacka nerodna namestitve; opomnimo naj, da je cilj diplomskega dela tečaj postavitve oblaka. Prepričan sem, da je izbira platforme OpenStack pravilna odločitev.

Platforma OpenStack omogoča virtualizacijo infrastrukture, na kateri tečejo



navidezni stroji, ki jih uporabljamo kot navadne računalnike z ustrezno nameščeno programsko opremo. Tipične potrebe vsakega računalniško usmerjenega podjetja pri poslovanju vključujejo namenske aplikacije, podatkovna skladišča, razvojna okolja, elektronsko pošto, organizacijska orodja, možnost ponujanja razvojnih okolij zunanjim izvajalcem in še mnoge druge. Pri tradicionalnem pristopu imajo podjetja programsko opremo nameščeno ločeno na več fizičnih strežnikov, s pomočjo platforme pa lahko fizične strežnike nadomestijo z navideznimi strežniki; tako iztisnejo zmogljivosti obstoječe infrastrukture in prihranijo na stroških vzdrževanja, ki so še vedno prisotni, vendar v manjšem obsegu.

## 1.3 Struktura diplomskega dela

V prvem poglavju sta predstavljena motivacija in namen diplomskega dela. V naslednjem poglavju iščemo smernice za izdelavo e-tečaja glede na psihologijo učenja v sistemih za e-izobraževanje. Sledi obširen opis računalništva v oblaku; najprej se seznanimo z zgodovinskim razvojem, nato opredelimo njegove značilnosti, ki postavijo mejo med podobnimi pristopi konsolidacije infrastrukture, nato si bomo pogledali storitve in nadzorne modele. Virtualizacija je sestavni del računalništva v oblaku. Njene zmožnosti so opisane v četrtem poglavju. Kako deluje platforma OpenStack, je podano v petem poglavju. Postopkom njene postavitve bomo obravnavali tekom sklopa vodenih praktičnih nalog v šestem poglavju. Z namenom preverjanja kakovosti in težavnosti tečaja smo opravili njegovo evalvacijo s strani manjše skupine učencev; raziskava je predstavljena v sedem poglavju. V zadnjem poglavju sta zaključek in opis možnosti za nadaljnjo uporabo oziroma nadgradnjo e-tečaja. V dodatku so slike, sheme platforme, opisi API-jev in vprašalnika.



## Poglavje 2

# Učenje v e-izobraževanju

Nove informacijsko-komunikacijske tehnologije pomembno vplivajo na vsa področja našega življenja, tudi na področje izobraževanja, ki je v osnovi še vedno enako, vendar se podaja v skladu z zmogljivostjo tehnologije. V okviru tega poglavja bomo zastavili tečaj, ki bo učencem omogočil postavitev platforme OpenStack. Zato se moramo najprej seznaniti z dejavniki in s pogoji, ki vplivajo na učenje; prav tako moramo poiskati smernice za vsebinsko kakovosten in primerno oblikovan tečaj.

### 2.1 Psihološke študije in uspešnost učenja

V družbi so asociacije o učenju večinoma povezane s šolskim učenjem: s pridobivanjem znanja, z branjem knjig in s ponavljanjem snovi. Zato je naša predstava ozko vezana na šolsko učenje ob knjigi in nas pri tem spremlja čustveno neugodje, ki ima slab vpliv na motivacijo. Definicija učenja je [44]:

*Učenje je vsaka sprememba v vedenju, informiranosti, znanju, razumevanju, stališčih, spretnostih ali zmožnostih, ki je trajna in ki je ne moremo pripisati fizični rasti ali razvoju podedovanih vedenjskih vzorcev.*

Povzetek številnih študij zatrjuje, da ima učenje kakovosten in količinski vpliv [44]:

- učenje je kvantitativno povečevanje znanja (količinski);
- učenje je sposobnost pomnjenja informacij z namenom kasnejšega priključevanja (količinski);
- učenje je trajno ohranjanje postopkov, metod in dejstev (količinski);
- učenje je proces ustvarjanja lastne razlage z namenom razumevanja resničnosti, življenja in sebe (kakovostni);
- učenje je proces, v katerem spreminjamo sami sebe in tako drugače gledamo na sebe ter pojave v svetu (kakovostni).

V procesu pridobivanja znanja ima vsak posameznik svoj stil spoznavanja, učenja in zaznavanja. Stili so izraz širših razsežnosti osebnostnega funkcioniranja in čustveno-motivacijske plati posameznika. Stil spoznavanja je dosledna in trajna posebnost posameznika v sprejemanju, ohranjanju, predelovanju in organiziranju informacij ter ga je mogoče opaziti pri vsakem posamezniku v situaciji soočanja s problemom. Spoznani stil vpliva na način in uspešnost reševanja problemov ter je razdeljen na naslednje stile: impulzivni, refleksivni, intuitivni ter analitični stil. Osebe, pri katerih prevladuje impulzivni stil, se odločijo za prvo možno rešitev, tudi če je le-ta napačna. Osebe z refleksivnim stilom preden se odločijo o reševanju problemov, premislijo o možnih rešitvah. Intuitivni se odločijo na podlagi svoje intuicije, opazovanja in preteklih čustvenih izkušenj. Osebe iz zadnjega stila se odločijo na podlagi analitičnega pristopa reševanja problema, ki poteka tako, da posameznik problem razdeli na več manjših podproblemov, ki jih zaporedno in posamezno reši ter na koncu sestavi v celoto. Učni stil poleg prvin spoznavnega stila vključuje še strategije učenja in doseganja ciljev. V zadnjem času se je na podlagi nevro-lingvističnega programiranja uveljavil stil zaznavanja, ki mu posameznik daje prednost pri sprejemanju in notranji predstavi učnih vtisov iz okolja. Tako ločimo tri zaznavne stile, in sicer [44]:

- vizualni stil (vidni);
- avditivni stil (slušni);
- kinestetični stil (gibalni).

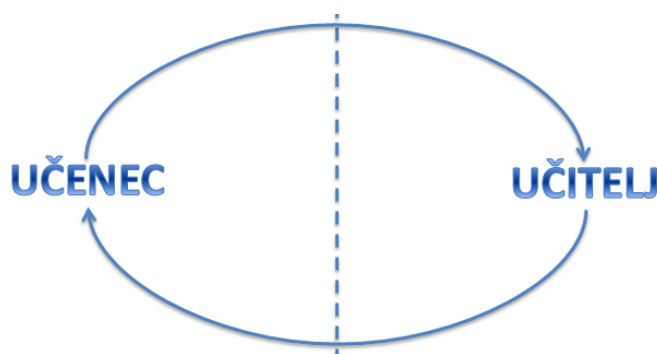
Pri vizualnem stilu si zapomnimo največ informacij na podlagi njihove vizualizacije. Za razumevanje in dolgotrajnejše ohranjanje informacije si moramo najprej ustvariti dobro vizualno predstavo s pomočjo slikovnih gradiv, skice in ustvarjanja lastnih miselnih vzorcev. Pri učenju smo organizirani in sistematični, učno snov si uredimo z barvami in jo raje beremo kot poslušamo. Pri avditivnem stilu si informacijo najbolje zapomnimo na podlagi zvoka. Zato pri učenju beremo na glas, se pogovarjamo sami s sabo, za utrjevanje učne snovi se o njej pogovarjamo in razpravljamo. Pristnost hrupa je zelo moteč dejavnik. Pri kinestetičnem stilu pa si informacijo najbolje zapomnimo, če smo v nenehni fizični aktivnosti. Pri učenju se nanašamo na intuicijo, fizično interakcijo s predmetom, izvajane gestikulacij med govorom in učenjem. Informacije si najbolje zapomnimo, če imamo usklajeno fizično aktivnost in miselni tok. Več nam pomeni celotna izkušnja kot podrobnosti. V procesu učenja so zastopani vsi trije stili; eden izmed njih prevladuje, ostala dva ga dopolnjujeta. Zato ne moremo trditi, da izključno uporabljamo samo enega [44].

Učna uspešnost učenca je odvisna od med sabo prepletenih notranjih in zunanjih dejavnikov. Notranji dejavniki so v samem učencu. Le-ti so fiziološke in psihološke narave. Fiziološki dejavniki vključujejo stanje čutil, zdravstveno stanje in senzomotorično koordinacijo. Psihološki dejavniki pa vključujejo kognitivne sposobnosti, stil učenja in spoznavanja, motivacijo, prirojene in privzgojene čustveno-vedenjske značilnosti, predznanje. Zunanji dejavniki so fizikalne lastnosti, kot sta opremljenost in oblikovanost učnega prostora, in številni socialni dejavniki, ki izvirajo iz družbenega okolja, kot je socialno-ekonomski status družine, podpora bližnjih, metode in sistem izobraževanja, učiteljeve sposobnosti poučevanja, izven izobraževalne dejavnosti, interakcija med učiteljem in učencem, pravila vedenja in komunikacije ter

vpliv sovrstnikov [44].

## 2.2 Klasičen pristop in e-izobraževanje

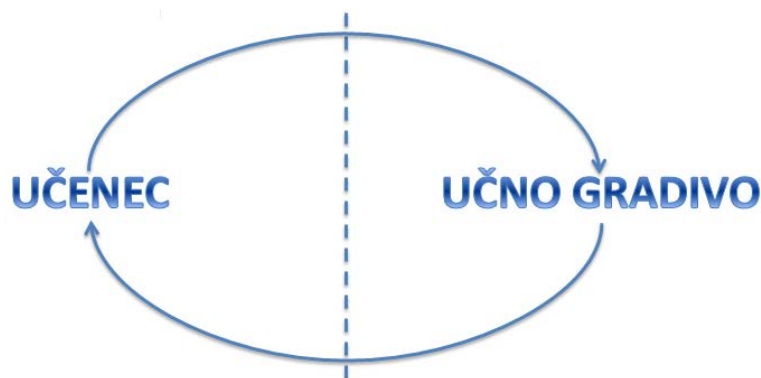
V klasičnem pristopu k poučevanju, ki je predstavljen na Sliki 2.1, učna interakcija poteka neposredno med učiteljem in učencem. Vloga učitelja je priprava učnega načrta, izvajanje potrebne dejavnosti za osvojitev znanja, prilagajanje učne dejavnosti glede na sposobnosti učencev (določeno tematiko se dlje časa uči in ponavlja), ovrednotenje učencev (testi) in vzdrževanje motivacije. V procesu poučevanja učitelj daje poudarek na množici povezanih podatkov, dejstev, domnev in teorije v okviru priporočenega učnega načrta ter ozkega pogleda na zanimanje učencev. Na učence jih prenaša s predavanji in z učbeniki. Učitelj pričakuje in zahteva, da učenci vsebino sprejmejo in jo pomnijo ter reproducirajo kot dokaz, da so se naučili. Vloga učenca je pri klasičnem načinu večinoma pasivna. Posledice tega so na daljši rok vidne v manjši učinkovitosti učenca, v manjši meri koristnega in trajnega znanja, motivacija učenca lahko upade. Posledice lahko privedejo do njegove osebne stiske, kar pogosto vodi k odvratanju od nadaljnjega izobraževanja [19, 68].



Slika 2.1: Interakcija pri klasičnem učenju med učiteljem in učencem.

V e-izobraževanju, ki ga imenujemo tudi e-učenje oziroma izobraževanje na

daljavo, je način izobraževanja, usposabljanja in izpopolnjevanja znanja, ki poteka s pomočjo sodobnih informacijsko-komunikacijskih tehnologij in sistemov za upravljanje izobraževanja (*ang. learning management system*). V takšnih sistemih se vzpostavi interakcija med učencem in učnim gradivom, kot je prikazano na Sliki 2.2 [19]. Večina opravil učitelja se pri klasičnem poučevanju prenese na samega učenca; vloga učitelja je v pripravi primerne gradiva, usmerjanja in tutorstvu. V takšnem učnem procesu učenec razvija svoje kognitivne in organizacijske sposobnosti, vsebine izbere v skladu s svojim zanimanjem, priuči se učinkovitega učenja, razvija sposobnost identifikacije problema in pristop k njegovemu reševanju, kritičnemu razmišljanju in selekciji med bistvenimi ter postranskimi informacijami, kar spodbuja k sposobnosti samostojnega učenja. Samostojno učenje je sposobnost samostojnega uravnavanja in nadzorovanja učenja ter sprejemanja pomembnih odločitev o poteku učenju [19, 68].



Slika 2.2: Interakcija med učencem in učnim gradivom v LMS-ju.

Rezultati novejših študij so pokazali, da ima vpliv na učno uspešnost prisotnost informacijsko-komunikacijske tehnologije, saj je le-ta za učenca vznemirljiva, spodbuja ga k uporabi in raziskovanju ter s tem tudi k učenju in izobraževanju. V veliki meri, tako klasičnega kot e-izobraževanja, je uspešnost učenja pogojena z motivacijo [67].

Motivacija je psihološki element posameznika, ki ga usmerja proti zastavlje-

nemu cilju. Je proces, ki določa intenzivnost, trajanje in kakovost določenega opravila z namenom doseganja boljšega rezultata. Glede na izvor motivacijo ločimo na zunanjo in notranjo. Pri zunanji motivaciji se učimo zaradi zunanjih posledic, kot so dobra ocena v šoli, pohvala ali graja s strani nadrejenega, pridobitev nagrade ali kazni, napredovanje na delovnem mestu in uveljavljanje v družbi. Samo dejavnost nas ne zanima preveč, temveč služi kot sredstvo za doseganje pozitivnih oziroma izogibanje negativnim posledicam. Pri notranji motivaciji smo dejavni zaradi želje po razvijanju lastnih sposobnosti, doseganja določenih spretnosti, radovednosti po spoznavanju in razumevanju nečesa novega. Doseganje zastavljenega cilja ni toliko pomembno kot doseganje lastnega zadovoljstva. Prednost notranje motivacije je poleg zadovoljstva tudi trajnost, saj nismo odvisni od zunanjih motivatorjev, ki jih okolica ne zmore vedno ustvariti [44].

## 2.3 Sistemi za upravljanje e-izobraževanj

Sistem za upravljanje izobraževanja (*ang. Learning Management System – LMS*) ali navidezno učno okolje (*ang. Virtual Learning Environment – VLE*) je programski paket, ki nam omogoča e-izobraževanje in ponuja različno vrsto gradiva. Večina takšnih sistemov temelji na uveljavljenih spletnih tehnologijah, ki se izvajajo na strežniku, kot so PHP [63], strežnik Apache [6] in podatkovna baza MySQL [49]. Pri tem uporabnik pošilja zahteve na strežnik, ki jo obdela in pošlje odgovor nazaj uporabniku. Ne obstaja nobena uradna definicija, ki bi natančno definirala, kaj takšen sistem mora vsebovati. V osnovi vsebuje in omogoča [19]:

- večji uporabniški pogled (skrbnik sistema, učitelj, učenec);
- administracijo ter nadgrajevanje sistema (uporabniške vloge);
- upravljanje z vsemi vrstami učnih gradiv;
- komunikacijo med učitelji in učenci;



- koledar in orodja za načrtovanje dela učiteljev ter učencev;
- sistem za ocenjevanje in spremljanje napredka učencev;
- forume in wiki strani [79].

E-izobraževanje ima za učenca številne slabosti in prednosti. Največja slabost je v pomanjkanju osebnega stika in komunikaciji z učiteljem ter s sodelujočimi učenci. Na drugi strani ima številne prednosti pred tradicionalnim pristopom poučevanja, kot so [18]:

- možnost prilagajanja lastnostim učenca (tip učenja, starost, računalniška pismenost, prejšnje izkušnje s sistemom, spol, materni jezik);
- način komunikacije med učencem in sistemom, ki je lahko intuitiven, zagotavlja dobro uporabniško izkušnjo in podpira GUI;
- podpora multimedijskemu gradivu, ki omogoča boljše razumevanje, pregled in strukturo gradiva (video vodiči, zvočni zapisi, uporabo barv in shem v učni vsebini);
- več načinov preizkusov znanja (vprašanj tipa DA/NE, možnosti izbire več odgovorov, odprte odgovore, vprašanja, podprta s slikami in z video vsebinami);
- možnost sledenja napredka in uspešnosti učenca.

## 2.4 Priprava tečaja

Zastavili smo si cilj izdelave spletnega tečaja, v katerem se bo učenec naučil postaviti platformo OpenStack. Pravilna zasnova spletnega tečaja je bistvenega pomena za kakovosten in celosten tečaj. Prvi korak k temu je opredelitev namena tečaja in udeležencev. Tečaj mora podajati vsa potrebna gradiva in informacije, s katerimi bo učenec po opravljenem tečaju pridobil znanje in razumevanje ter bo sposoben samostojno postaviti platformo. V osnovi bo

namenjen neizkušenim skrbnikom sistemov in računalničarjem, še vedno pa mora podajati smernice za nadaljnje samostojno učenje. K temu bomo ustrezno prilagodili stopnjo težavnosti tečaja in zastavili primerno sistematično učno pot. Učna pot bo obsegala več učnih enot, ki jih bo učenec moral obravnavati zaporedno [67]. Ker se bo tečaj izvajal v okviru e-izobraževanja, je potrebno upoštevati smernice pri razvoju takšnega okolja. V le-tega mora biti zajeto [44]:

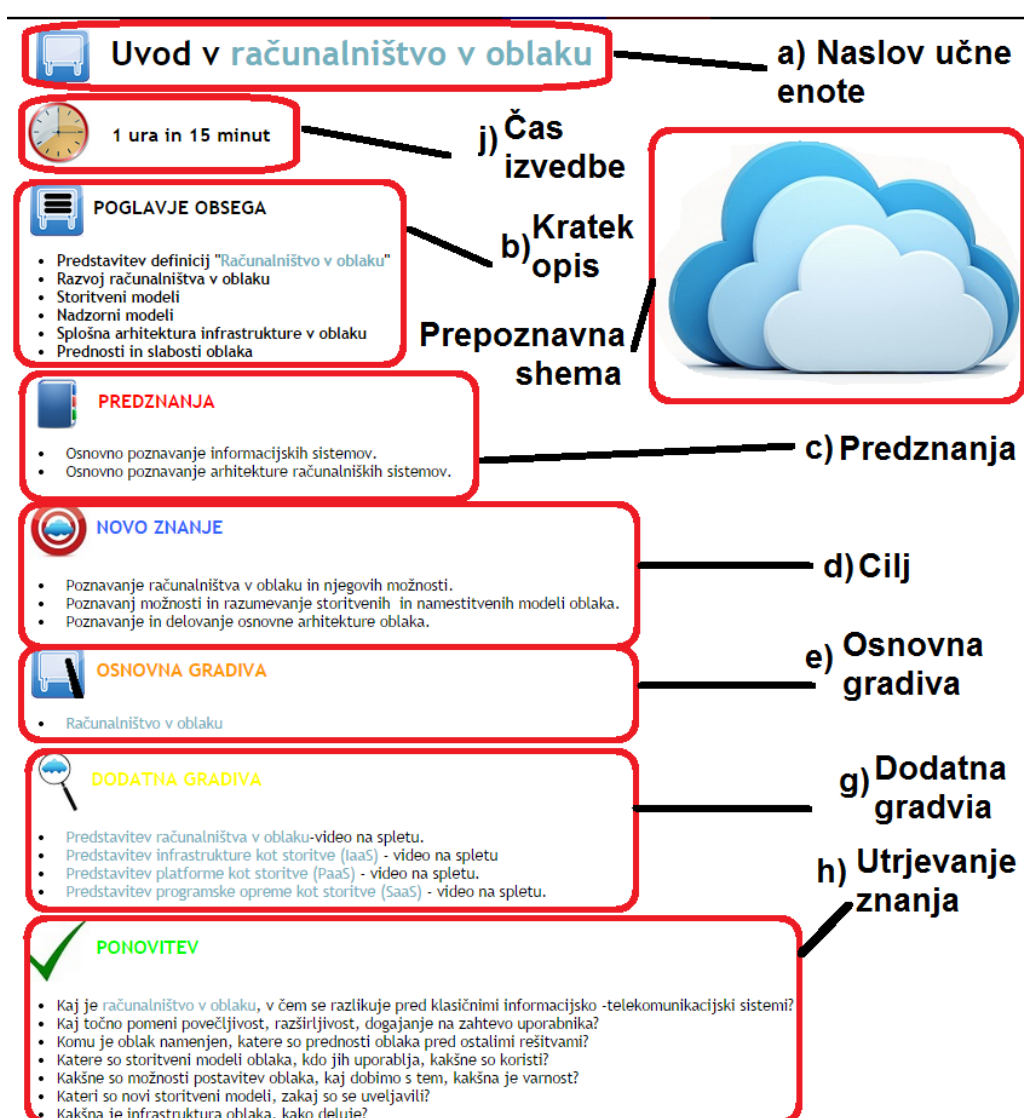
- postopkovno voden učni proces, ki ima jasno določen cilj;
- priprava gradiv, tako za učence s slabim predznanjem kot za učence z zanimanjem po dodatnih in obširnejših znanjih;
- vključenost dodatnih in multimedijskih gradiv, ki spodbudijo motivacijo ter zanimanje;
- ustrezen prehod z osnovnega nivoja na zahtevnejši nivo tematik;
- obojestransko aktivno sodelovanje med učencem in učiteljem ter možnost spremljanja napredka učenca;
- prostor, v katerem bi lahko učenci izmenjevali informacije in izkušnje, zastavljali vprašanja in prejeli odgovore;
- učno okolje z grafičnim uporabniškim vmesnikom, ki pripomore k dobri uporabniški izkušnji učenca;
- možnost povratnih informacij o primernosti vsebin, poteku tečaja, primerni zastavljenosti in oblikovanosti tečaja.

Glede na lastnosti dobro zasnovanega e-učnega okolja smo se odločili, da bomo tečaj zastavili v spletni učilnici Moodle (*ang. Modular Object-Oriented Dynamic Learning Environment - MOODLE*), ki se nahaja na naslovu <http://rtk2014.fri.uni-lj.si/>. Tečaj mora omogočiti učenje po korakih; učencem, ki določeno gradivo že poznajo, mora omogočiti preskok na naslednje gradivo. Zato je potrebno, da se tečaj razdeli na več učnih enot. V okviru tečaja morajo biti zastavljene različne aktivnosti, kot so forumi in klepetalnica, ki

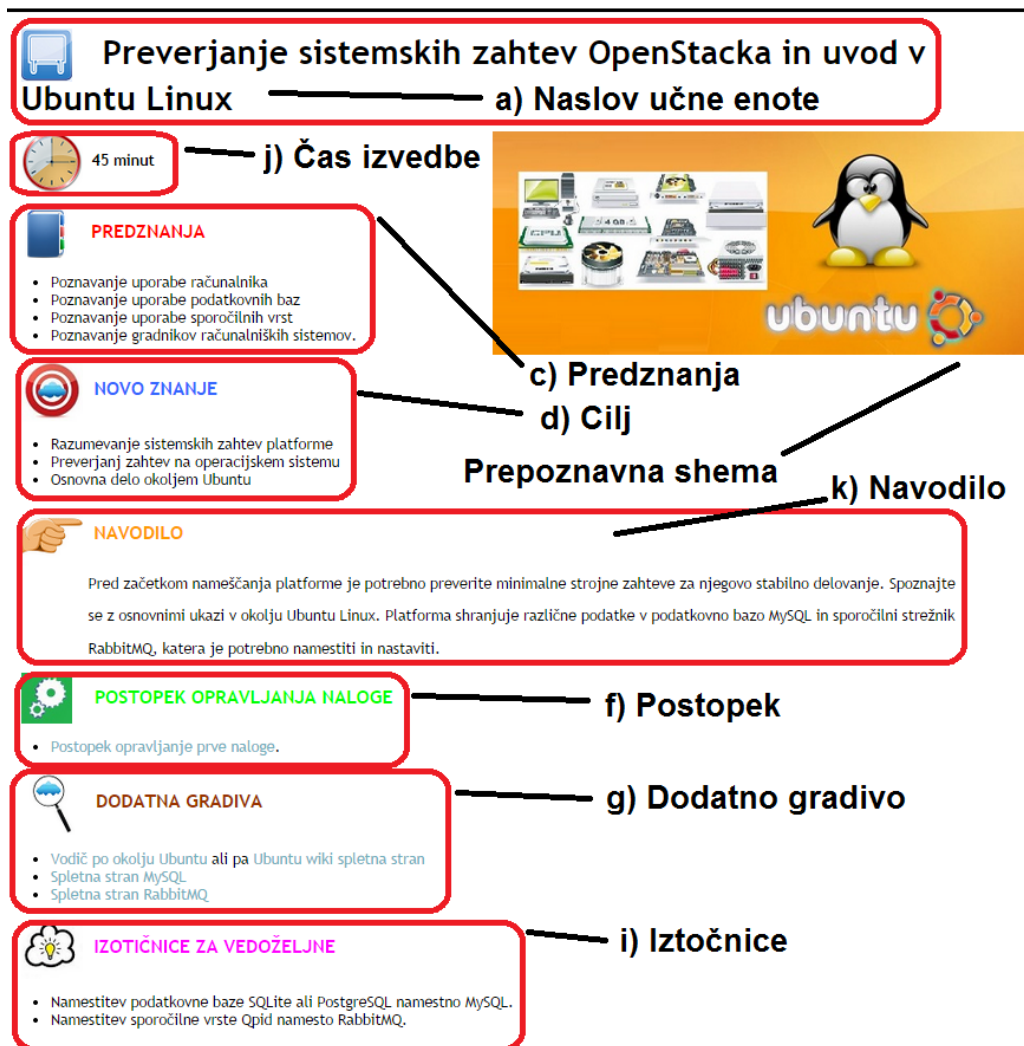
povečujejo aktivno medsebojno sodelovanje učencev. Vsebina, ki bo učencem podana, nikakor ne sme biti pretežka za razumevanje, biti mora zapisana v strokovnem računalniško-tehničnem slovenskem jeziku, vendar še vedno dovolj razumljiva in vsebinsko popolna. Vsaka učna enota mora biti sistematično razdeljena, pregledna in intuitivna. Glede na cilj tečaja in značilnosti učencev je učno pot potrebno uvesti v teoretični ter praktični del tečaja, ki se med sabo dopolnjujeta. Tako teoretični kot praktični del tečaja bo zasnovan na posameznih učnih enotah, kot je prikazano na Slikah 2.3 in 2.4. Vsaka učna enota bo vsebovala:

- naslov učne enote in (Slika 2.3, a);
- kratek opis učne enote, v kateri je strnjena vsebina učne enote (Slika 2.3, b);
- predznanja, ki jih mora imeti učenec pred začetkom učenja učne enote (Slika 2.3, c);
- cilj, znanja učenca, ki jih bo pridobil po opravljeni učni enoti (Slika 2.3, d);
- osnovna gradiva, ki jih učenec mora samostojno predelati po korakih (Slika 2.3, e);
- postopek izvedbe naloge, učencu je tekom reševanja praktične naloge v pomoč natančen postopek (Slika 2.4, f);
- dodatna gradiva, kot so datoteke in sheme za učenje (Slika 2.5, g);
- vprašanja za ponovitev in utrjevanje znanja v učni enoti (Slika 2.3, h);
- iztočnice, za učence, ki se želijo poglobiti in nadgraditi svoje znanje, preveriti dodatne možnosti postavitve ter namestitve platforme (Slika 2.3, i);
- čas izvedbe, ki ga potrebuje povprečni učenec, da predela osnovna učna gradiva (Slika 2.3, j);

- prepoznavno shemo učne enote (Slika 2.3, k).



Slika 2.3: Teoretična učna enota *Uvod v računalništvo v oblaku*.



Slika 2.4: Praktična učna enota *Preverjanje sistemskih zahtev OpenStacka in uvod v Linux Ubuntu*.

Poleg prvin pisanja učnih enot je v praktični učni enoti potrebno dodati kakovostna, strokovno pravilna in nedvoumno napisana navodila za reševanje praktičnih nalog, ki bodo vsebovala številne razlage tekom postopka izvedbe vaje. Vsaka praktična učna enota je razdeljena na več podenot, ki so logično povezane in oblikovane. Vsaka podenota ima svoj naslov; izvaja se zaporedno in vsebuje kratka navodila, nastavitve ter dodatna pojasnila v obliki

komentarjev. Kjer je potrebno biti pazljiv oziroma si je potrebno zabeležiti uporabniške podatke, so podane grafične ikone za dodatno nazornost. Na koncu enote so ukazi, s katerimi preverimo delovanje storitev; v nasprotnem primeru smo napačno nastavili neko nastavitvev. Preverjanje se izvede s ponovnim zagonom storitve, s preverjanjem izvršljivosti nekega ukaza in povratno informacijo. Primer oblikovanja navodil v praktični učni enoti je predstavljen v nadaljevanju.

**NASLOV PODENOTE:** Namestitev in nastavitve računske komponente

**KRATKO NAVODILO:** Najprej je potrebno namestiti vse potrebne procese Nove, nato ji omogočiti dostop do baze in jo ustrezno nastaviti:

```
$ sudo apt-get install nova-api nova-cert nova-conductor
```

```
nova-consoleauth nova-novncproxy nova-scheduler
```

```
python-novaclient nova-compute-kvm nova-network
```

V nastavitveni datoteki `/etc/nova/nova.conf` ustvarimo razdelek `[database]` in nastavimo povezavo na bazo.

**NASTAVITEV:**

```
[database]
```

```
connection=mysql://novaUser:novaDBPass@<ime_našega_računalnika>/
```

```
novaDatabase
```

**GRAFIČNA SIMBOL ZA DODATNO NAZORNOST**



Pri tem si moramo zapomniti uporabniško ime in geslo, ki ga bomo nastavili.

V enaki nastavitveni datoteki je računskim storitvam potrebno nastaviti tudi dostop do sporočilnega strežnika RabbitMQ. Pod razdelek `[DEFAULT]` je potrebno namestiti kateri sporočilni strežnik se uporablja, kje se nahaja in

geslo za dostop.

#### DODATNA POJASNILA V KOMENTARJIH:

```
rpc_backend = rabbit //za sporočilni strežnik nastavimo rabbit
rabbit_host = <ime_našega_strežnika> //nastavimo gostitelja strežnika
rabbit_password = rabbitpass //za dostop storitve do sporočilnega strežnika
//moramo nastaviti geslo
```

V enaki nastavitveni datoteki je potrebno nastaviti IP naslove, na katerih se lahko izvajajo storitve oddaljenega dostopa do komponente:

```
[DEFAULT]
my_ip = 192.168.200.26 //nastavimo naslov IP našega računalnika
vncserver_listen = 192.168.200.26 //nastavim naslov IP kjer lahko poteka
//oddaljeni dostop
vncserver_proxyclient_address = 192.168.200.26 //nastavimo naslov IP
//proxy strežnika
vnc_enabled = True //omogočimo oddaljeni dostop
novncproxy_base_url = http://<ime_našega_strežnika>:6080/vnc_auto.html
glance_host = <ime_našega_strežnika> //nastavimo ime gostitelja
//komponente Glance
```



Pri nastavljanju moramo biti pozorni na to, da nastavimo svoj IP naslov.

V nastavitveno datoteko pod razdelek *[DEFAULT]* je potrebno nastaviti, da se bo storitev preko storitve Keystone avtenticirala:

```
[DEFAULT]
```

*auth\_strategy = keystone //storitev se avtenticira preko storitve Keystone*

Nato ustvarimo razdelek *[keystone\_authtoken]*, kamor vnesemo prijavne podatke storitve. Opisa parametrov sta enaka prejšnjemu.

*[keystone\_authtoken]*

*auth\_uri = http://<ime\_našega\_strežnika>:5000*

*auth\_host = ime\_našega\_strežnika>*

*auth\_port = 35357*

*auth\_protocol = http*

*admin\_tenant\_name = service*

*admin\_user = novaService*

*admin\_password = novaPass*

Potem še v nastavitvenih datotekah preverimo pravilnost nastavljenih parametrov. Če se vsi procesi ponovno zaženejo, smo jih pravilno nastavili:

*\$ sudo service nova-api restart*

*\$ sudo service nova-cert restart*

*\$ sudo service nova-consoleauth restart*

*\$ sudo service nova-scheduler restart*

*\$ sudo service nova-conductor restart*

*\$ sudo service nova-novncproxy restart*

*\$ sudo service nova-compute restart*

*\$ sudo service nova-network*



V kolikor se kateri izmed procesov ne zažene, ste opravili napako v nastavitvah, zato še enkrat preverite nastavitvev.



### Celoten pregled tečaja

V prvem delu je kratek povzetek tečaja, ki opisuje potrebna znanja učencev, znanja, ki jih bodo učenci dobili tekom tečaja, in ciljna skupina tečaja. Prve tri učne enote so teoretične. V prvi učni enoti učenci osvojijo znanje računalništva v oblaku. V drugi se seznanijo z virtualizacijo in virtualizacijskimi tehnikami. V tretji učni enoti se spoznajo s platformo OpenStack. Naslednje učne enote so praktične. V praktičnem delu pa bomo učencem omogočili podpreti osvojena teoretična znanja s postavljanjem platforme OpenStack na OS Ubuntu Linux. Praktični del bomo razdelili na štiri učne enote. V prvi jih bomo seznanili s sistematskimi zahtevami platforme, v drugi bomo nastavili in namestili komponente, v tretji bomo vzpostavili okolje, v četrti enoti pa bomo pognali navidezni stroj. Praktičen del tečaja je zastavljen zaporedno in posameznih učnih enot se ne more preskočiti. V kolikor učenec uspešno opravi praktično učno enoto, bo lahko začel z naslednjo.

Pri oblikovanosti tečaja je potrebno vpeljati prvine zaznavnih stilov, ki smo jih obravnavali na začetku tega poglavja. Tako bodo učenci učinkovito in v zglednem času osvojili znanja za samostojno postavitve tečaja. Elementi v tečaju, ki podpirajo vizualno zaznavo, so:

- prirejeni naslovi poglavij, vsak ima svojo ikono;
- vsaka učna enota ima svojo prepoznavno shemo
- vsaka učna enota ima poleg razlag tudi številke slike in sheme;
- med reševanje praktičnih učnih enot učenec vidi, kaj se dogaja na zaslonu.

Elementi v tečaju, ki podpirajo avditivno zaznavo, so:

- med reševanjem praktičnih učnih enot učenec lahko sliši varnostna opozorila operacijskega sistema;
- postopek si lahko bere na glas.

Elementi v tečaju, ki podpirajo kinestetično zaznavo, so:

- v vsaki učni enoti je neko praktično delo;
- učenec pri praktični učni enoti pretipkava ukaze, opazuje dogajanje na zaslonu in prejema povratne informacije.

V učnih enotah se uporabljajo številni simboli, ki izboljšajo grafični vmesnik in posledično vizualno zaznavo ter boljšo uporabniško izkušnjo. Tako pripomorejo k lažjemu učenju in k večjemu pregledu strani.



**Naslov učne enote** – vsaka učna enota ima svoj naslov; tako učenec lažje loči med posameznimi poglavji in drsi po spletni strani (Slika 2.3 a, slika 2.4 a). Simbol nepopisane table ponazarja naslov enote.



**Kratek opis učne enote**, v kateri je strnjena vsebina posamezne učne enote (Slika 2.3 b). Simbol table s črtami predstavlja povzetek enote.



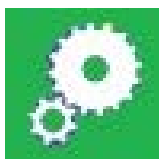
**Predznanja**, ki jih mora imeti učenec pred začetkom izvajanja učne enote (Slika 2.3 c, Slika 2.4 c). Simbol zaprte knjige predstavlja, da je učenec knjigo zaprl, ko se je določeno tematiko v preteklosti naučil.



**Tarča** – cilji in znanja, ki jih učenec pridobi po opravljeni učni enoti (Slika 2.3 d, Slika 2.4 d). Simbol tarče, ki jo splošno povezujemo s cilji.



**Osnovna gradiva**, ki jih učenec mora samostojno predelati po korakih (Slika 2.3 e). Simbol table z učiteljevo palico, ki zahteva, kaj se je potrebno naučiti.



**Postopek** – natančen postopek izvedbe naloge, ki je na voljo učencu tekom reševanja praktične učne enote (Slika 2.4 f). Simbol mehanskega pogonskega sklopa, ki splošno ponazarja delovanje nekega sistema.



**Dodatna gradiva** – dodatne informacije o tečaju, dodatna gradiva in sheme za učenje (Slika 2.3 g, Slika 2.4 g). Simbol lupe splošno ponazarja natančno iskanje dodatnih informacij.



**Utrjevanje** – točke za ponovitev in utrjevanje znanja v učni enoti (Slika 2.3 h). Simbol kljukice v splošnem predstavlja, da smo opravilo uspešno opravili.



**Iztočnice** – za učence, ki želijo nadgraditi svoje znanje, preveriti dodatne možnosti postavitve in namestitve platforme (Slika 2.4 i). Simbol vklopljene luči na splošno povezujemo z ustvarjanjem idej.



**Čas izvedbe** – čas, ki ga potrebuje povprečni učenec, da predela učno enoto (Slika 2.3 j, Slika 2.4 j). Simbol ure v splošnem predstavlja porabo časa.



**Beležka** – v nekaterih učni enotah si je potrebno zapomniti pomembne informacije, kot so uporabniško ime in gesla. Simbol beležke na splošno predstavlja potrebo po beleženju neke informacije.



**Opozorila** – v nekaterih učnih enotah je potrebno biti pozoren tudi na malenkosti. Simbol klicaj na splošno predstavlja previdnost.



**Navodilo** – opisuje, kaj je potrebno opraviti pri praktični učni enoti (Slika 2.4 k). Simbol kazalca v splošnem pomeni usmerjanje pozornosti učenca.

## Poglavje 3

# Računalništvo v oblaku

V tem poglavju bomo predstavili ozadje razvoja računalništva v oblaku, njegove bistvene lastnosti, s katerimi ga je moč prepoznati pred drugimi rešitvami v informacijsko- komunikacijskih tehnologijah. Nato bomo razložili storitvene in namestitvene modele ter prednosti in slabosti uporabe za podjetja.

### 3.1 Razvoj računalništva v oblaku

Računalništvo v oblaku v resnici ne predstavlja nobene revolucionarne tehnologije oziroma njenega napredka. V bistvu gre za združitev že obstoječih in uveljavljenih tehnologij, ki je uporabniku predstavljeno nekoliko drugače. Strokovni izraz računalništvo v oblaku izhaja iz shematske upodobitve interneta, ki se ga po navadi upodobi z oblakom. Shema oblaka je preprosta in ne podaja nobenih podrobnosti, saj takšen vtis želi končni uporabnik, ki mu je prioriteta dosegljivost aplikacij in podatkov ne glede na čas ter lokacijo dostopa. V resnici gre za zapleteno infrastrukturo, ki jo tvorijo gruče strežnikov, komunikacijskih naprav, diskov in zapletena programska oprema. Preboj določene tehnologije na tržišče je odvisen predvsem od obstoječih tehnologij, njene uporabe, priljubljenosti in slabosti, smernic prihajajočih tehnologij,

ki so v razvoju, združevanja velikih podjetij z enakimi interesi, ekonomskih situacij in agresivnega oglaševanja ter celo vplivov kulturnih prepričanj. Večje spremembe na področju informacijsko-komunikacijskih tehnologij je mogoče zaznati na približno 10 let. K masovni uporabi računalništva v oblaku so bistveno pripomogle naslednji tehnologije: virtualizacija, spletne storitve in njihova arhitektura, spletne inačice 2.0 (Web 2.0) in širokopasovne internetne povezave. Virtualizacija omogoča boljšo izkoriščenost strojne opreme in zmanjšuje stroške vzdrževanja ter električnega napajanja. V naslednjem poglavju si jo bomo podrobneje pogledali. Storitve računalništva v oblaku so oblikovane po standardu WSDL [80], ki definira vmesnik spletne storitve, in SOAP [71], ki omogoča komunikacijo med aplikacijami, ki tečejo na različnih programskih jezikih in operacijskih sistemih, ter uporabo spletnih tehnologij za spletne strani, kot sta HTML [70] in PHP. Spletna inačica 2.0 omogoča združevanje različnih virov podatkov in prikaz uporabniku preko grafičnega vmesnika [11].

Prvi zametki tovrstnega računalništva segajo v petdeseta leta prejšnjega stoletja (1. faza), kjer se je pojavila uporaba osrednjih računalnikov (*ang. mainframe*), kot prikazuje Slika 3.1. Računske zmogljivosti, ki so bile končne, so bile izključno na voljo raziskovalnim in vojaškim organizacijam preko terminalov, ki so se povezali na osrednji računalnik ter so imeli vlogo uporabniškega vmesnika. Leta 1960 je John McCarthy, profesor Stanfordske Univerze, napovedal, da bodo v prihodnosti storitve računskih centrov na voljo kot javna storitev, kot so oskrba z električno energijo oziroma vodovodnim omrežjem, saj ima takšen poslovni model za uporabnika nizke stroške. Kasneje je leta 1966 Douglas F. Parkhill v svoji knjigi »*The Challenge of the Computer Utility*« že opredelil smernice razvoja računalniške industrije; poleg uporabe virov kot javnih storitev je dodal še možnost oddaljenega dostopa preko omrežja, ponujanje navidezno neskončnih virov in možnost elastičnosti [11].

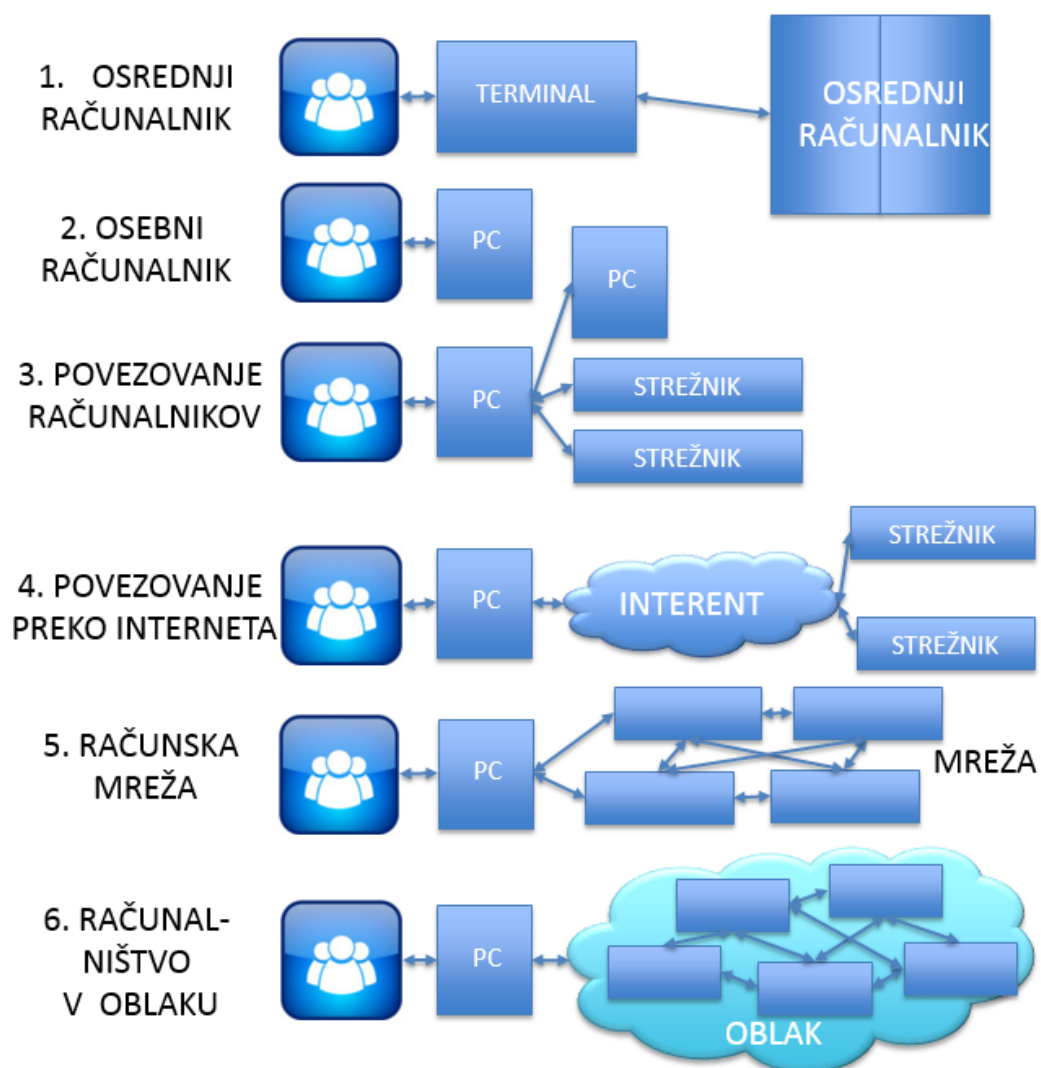
Tekom šestdesetih let (2. faza) sta se računska zmogljivost in količina po-

mnilnikov povečali in tako sta zadovoljili potrebo večine uporabnikov, kot je prikazano na Sliki 3.1. V osemdesetih letih (3. faza) je podjetje IBM [35] predstavilo prvi osebni računalnik, kot je prikazano na Sliki 3.1, in tako omogočilo njegovo globalno. V tem času se pojavi povezovanje osebnega računalnika, prenosnika in strežnika preko lokalnega omrežja ter tako omogoči delitev virov in doseganje hitrejšega delovanja. Na prehodu v devetdeseta leta (4. faza) se začne uporaba globalnega omrežja (interneta) in tako omogoči povezovanje med lokalnimi omrežji, prikazano na Sliki 3.1. V devetdesetih letih (5. faza) sta Ian Foster in Carl Kesselman predlagala, da bi dostop do računalniških virov moral biti enak kot povezava na omrežje električne energije. Računalnike so začeli povezovati v računske mreže (*ang. grid computing*), kot je prikazano na Sliki 3.1. Tako so uporabniki preko internetnih povezav dostopali do združenih računalniških virov. Ramnath K. Chellappa je leta 1997 je v pogovoru na srečanju INFORMS [38] (*ang. The Institute for Operations Research and the Management Science*) uporabil akademsko definicijo, v kateri je dejal, da je računalništvo v oblaku nov vzorec, kjer meje računalništva ne bodo postavljale tehnološke omejitve, temveč ekonomska načela. Nekatera podjetja so začela prepoznavati nove priložnosti. Prvo med njimi je bilo podjetje Salesforce.com, ki je poslovne aplikacije preko internetnega omrežja ponudilo uporabnikom. Po letu 2000 (6. faza) so številna računalniška podjetja začela vlagati v razvoj tehnologij, ki omogočajo storitve oblaka, kot so neskončna računska moč in shramba. Dostop je omogočen preko osebnega računalnika, ki ima prav tako računsko moč in možnost lokalne shrambe, vendar v precej manjšem obsegu kot oblak, prikazano na Sliki 3.1. Prve storitve so omogočale uporabo aplikacij. Ključno vlogo pri razširjanju oblačnih storitev je odigralo podjetje Amazon [2], ki je začelo posodabljati svoje podatkovne centre s prepričanjem, da lahko bolj izkoristijo svoje kapacitete. Leta 2002 je omenjeno podjetje omogočilo razvijalcem uporabo svojih aplikacijskih programskih vmesnikov (*ang. Application Programming Interface – API*). Leta 2004 se je pojavil splet različice 2.0 (*ang. web 2.0*), ki je prinesel spletne storitve (*ang. web services*). Takrat so podjetja začela

izdelovati aplikacije, ki so uporabljale lastne ali tuje javne storitve, ki so delovale znotraj brskalnikov [11].

Omenjeno podjetje je leta 2006 predstavilo storitve Amazon Web Services (AWS), ki omogoča Elastic Compute Cloud (EC2) [29] in Simple Storage Service (S3) [69]. Oba sta na voljo končnim uporabnikom v obliki navideznega stroja oziroma skladišča podatkov. Kasneje so se priključila ostala računalniška podjetja s svojimi vrstami storitev, kot so Google [33], IBM, HP [34] in Microsoft [45]. Leta 2008 sta se pojavili odprtokodni orodji OpenNebula [57] in Eucalyptus [30]; slednji je omogočil samo izdelavo zasebnega oblaka, prvi je še dodal možnost hibridnega oblaka. Leta 2010 sta Rackspace Hosting [66] in NASA [51] predstavila skupni projekt OpenStack, kot operacijski sistem za infrastrukturo oblaka. V istem letu je bilo na voljo tudi orodje CloudStack. V nadaljevanju diplomskega dela se bomo spoznali z lastnostmi in s storitvami oblaka.





Slika 3.1: Razvoj računalništva v oblaku skozi čas.

## 3.2 Opredelitev računalništva v oblaku

Računalništvo v oblaku je izraz, ki se danes pogosto pojavi v informacijsko-tehnološkem pogovoru. Kljub temu se predstave in razumevanje storitvenih modelov razlikujejo glede na tehnološko ozadje posameznika. V krogu skrbnikov in arhitektov sistemov IT je moč zaslediti dobro razumevanje konceptov,

ki jih prinaša računalništvo v oblaku, kot so: elastičnost, samopostrežba, razpoložljivost in povečljivost, vendar na drugi strani pa pozabljajo, da je še vedno osnovni gradnik povsem navaden strežnik. V širšem krogu programerjev je moč opaziti samo preprosto predstavo strežnika [75]. Oblak ponuja izključno storitve, ki so lahko skoraj kar koli. Pojem računalništvo v oblaku opredeljuje več definicij, ki so jih postavila različna podjetja in raziskovalne organizacije. Definicija Ameriškega nacionalnega inštituta za standarde in tehnologijo se glasi [62]:

*Računalništvo v oblaku je model, ki omogoča dostop do omrežja na zahtevo do deljene skupine računalniških virov (npr. omrežje, shramba podatkov, aplikacije in storitve), ki jih lahko hitro rezerviramo in sprostimo z minimalnim upravljalnim trudom oziroma z minimalnim posredovanjem ponudnika storitev. Model oblaka je sestavljen iz petih bistvenih karakteristik, treh storitvenih modelov in štirih načinov namestitve.*

Definicija družbe Gartner, vodilne družbe na področju raziskovanja informacijske tehnologije, se glasi [20]:

*Slog računalništva, kjer so masivno povečljive in elastične zmogljivosti IT dostavljene zunanjim uporabnikom kot storitev z uporabo internetnih tehnologij.*

Definicija agencije Forrester, neodvisne tehnološke in marketinške raziskovalne agencije, se glasi [21]:

*Standardizirane IT kapacitete (storitve, programske opreme ali infrastrukture), dostavljive na podlagi plačilnega modela na klik in samopostrežnega modela.*

S tehnološkega stališča je računalništvo v oblaku način zagotavljanja in upravljanja storitev IT, ki ima vse ali vsaj večino od naslednjih bistvenih lastnosti oblaka [62]:

- **samopostrežbo na zahtevo** (*ang. on-demand self-service*): uporabnik storitev v oblaku si lahko v skladu s svojimi potrebami sam zahteva računalniške vire oziroma kapacitete (čas uporabe strežnika, shrambe in podobno), brez potrebe po interakciji s strani ponudnika storitev;

- **širokopasovni omrežni dostop** (*ang. broad network access*): vsi viri so na voljo preko standardnih omrežij in omrežnih protokolov ter dosegljivi širokemu spektru uporabnikov in naprav;
- **združevanje (konsolidacija) virov** (*ang. resource pooling*): viri (procesor, pomnilnik, hramba podatkov, omrežje) so združeni v celoto, v kateri nastopajo fizični in navidezne viri, ki jih uporabniki zasežejo ali pa sprostijo. Pri tem ne vedo, na katerem fizičnem viru so dejansko shranjeni njihovi podatki oziramo na katerem viru se izvajajo njihovi procesi;
- **hitra elastičnost** (*ang. rapid elasticity*): uporabnik storitev v oblaku lahko hitro pridobi oziroma sprostí neuporabljene vire, ki jih oblak tako omogoči drugim uporabnikom, viri so uporabniku na voljo ne glede na čas in količino;
- **merjena storitev** (*ang. measured service*): sistem oblaka samodejno izvaja kontrolo in optimizacijo virov glede na vpliv uporabe. Uporabo virov v oblaku je mogoče meriti, nadzirati in beležiti, kar posledično omogoča transparentnost storitev tako na strani ponudnika kot uporabnika ter primerno zaračunavanje storitev po porabi (*ang. pay-per-use*).

### 3.3 Storitveni modeli računalništva v oblaku

Storitveni modeli določajo, kako se viri podredijo bistvenim lastnostim oblaka in kako viri sledijo storitvenemu modelu oblaka, kar na skladu storitev IT (*ang. IT stack*) omogoča postaviti mejno črto med klasičnim IT-pristopom in oblakom. Če povzamemo, oblak v osnovi ponuja naslednje storitvene modele [62], ki so prikazani na Sliki 3.2:

- infrastruktura kot storitev (*ang. Infrastructure as a Service – IaaS*);
- platforma kot storitev (*ang. Platforma as a Service – PaaS*);

- programska oprema kot storitev (*ang. Software as a Service – SaaS*).

### **Infrastruktura kot storitev**

Storitev ponuja vire, kot so procesorji, pomnilniki, shramba in omrežje, na katere uporabnik namesti platforme (operacijske sisteme, API-je), aplikacije in podatke ter tako vzpostavi delovno okolje, kot je prikazano na Sliki 3.2. Ponudnik upravlja oziroma kontrolira infrastrukturo oblaka. Primera sta Googlov Compute Engine [23] in Amazonov EC2. Storitev je namenjena podjetjem za lastne ali komercialne namene. Ponudnik mora zagotoviti nenehno dostopnost in razpoložljivost strojne opreme ter se pri tem ne vtika v opravila uporabnika, kar predstavlja večjo svobodo in posledično tudi več dela. Ponudnik storitev skrbi za upravljanje, vzdrževanje in nadgrajevanje infrastrukture, ki vključuje strežnike, shrambo podatkov ter dostopnost omrežja.

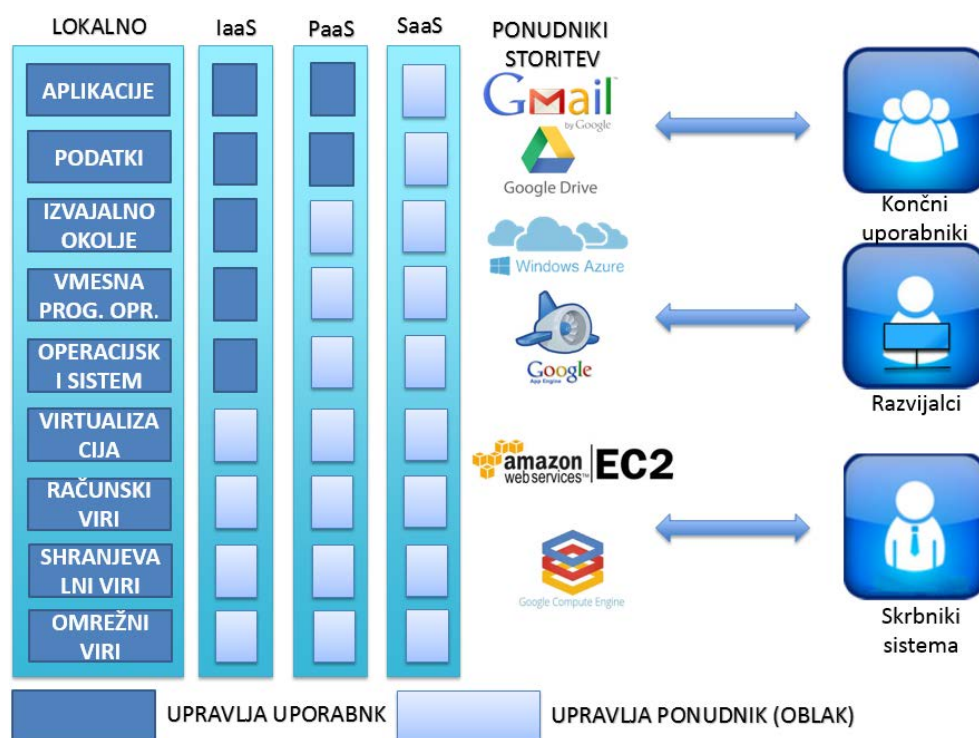
### **Platforma kot storitev**

Storitev združuje lastnosti zgornje storitve s storitvami, kot so operacijski sistemi, API-ji, izvajalna okolja programskih jezikov, procesi, knjižnice in ostala orodja, kot je prikazano na Sliki 3.2. Glede na ponujene storitve uporabnik skrbi za namestitev aplikacije, konfiguracijo in razvoj aplikacije ter za prenos podatkov. Ponudniku storitev se poleg zgoraj omenjenih zahtev pridružijo še zahteve po posodabljanju orodij, knjižnic in razvojnih okolij. Uporabnik pri tem plačuje uporabo platforme in nima več toliko svobode; posledično ima tudi manj dela. Primer sta Googlov App Engine [5] in Microsoftov Azure [10].

### **Programska oprema kot storitev**

Storitev ponovno združuje lastnosti predhodnih storitev z aplikacijami, ki se izvajajo na infrastrukturi ponudnika, kot prikazuje Slika 3.2. Uporabnik uporablja aplikacije na zahtevo (*ang. on-demand software*). Primer sta Googlova Gmail [32] in Drive [27]. Dostop do spletnih aplikacij poteka preko brskalnika, brez predhodno nameščenih aplikacij ali programskega vmesnika. Uporabnik samo prenese podatke na oblak, ki je povsem vezan na obstoječo aplikacijo,

katero ni mogoče prilagajati ali razvijati za lastne namene. Aplikacijo lahko uporablja brezplačno ali plačuje mesečno najemnino oziroma licenco. Ponudnik mora poleg zahtev, omenjenih v prejšnjih storitvah, zagotavljati še nenehno dostopnost in pravilno delovanje aplikacij, nadgrajevati programsko opremo, zagotavljati varnost in neokrnjenost uporabnikovih podatkov ali posege s strani tretjih oseb.



Slika 3.2: Osnovni storitveni modeli oblaka.

Z notraj istega oblaka je mogoče vzpostaviti kombinacijo pristopov IaaS, PaaS in SaaS [12].

### Novi storitveni modeli

Poleg predstavljenih modelov se zadnje čase uveljavljajo storitveni modeli, ki se zahtevam uporabnikom še bolj prilagodijo. Na voljo prihajajo posamezne komponente oblaka, ki so bile v preteklosti za uporabnika skrite ali pa jih je moral uporabljati v kombinaciji osnovnih storitvenih modelov in ustreznega

vmesnika. Le-ti so: dostop do aplikacij na zahtevo (*ang. Application as a Service – AaaS*), ki ga ponuja podjetje Venture Technologies [4], EMC [16] ponuja računske storitve za podatkovne centre (*ang. Computing as a service – CaaS*), AT&T ponuja storitev hrambe Synaptic cloud storage service (*ang. Storage as a Service – StaaS*) [72] in podjetje Pacnet ponuja omrežje kot storitve (*ang. Network as a Service – NaaS*) [52] ter druge, ki jih ne bomo omenjali.

### 3.4 Nadzorni modeli računalništva v oblak

Pri računalništvu v oblaku nastopata dva udeleženca, in sicer uporabnik ter ponudnik. Ponudnik virov ponuja storitve, ki delujejo po storitvenih modelih. Uporabnik izkorišča vire za svoje sisteme, aplikacije in podatke (IaaS), aplikacije s podatki (PaaS) ali pa samo podatke (SaaS). Ponudnik ima lahko v vsakem trenutku vpogled v delovanje in upravljanje virov, teoretično dostop do vseh podatkov na virih (ni nujno pravno dovoljeno) in pravico odločanja o uporabi teh virov. Postavitveni ali nadzorni modeli določajo, kdo je ponudnik virov in uporabnik virov. Glede na oba udeleženca ločimo pristope [12], ki so prikazani na Sliki 3.3. Le-ti so:

- zasebni oblak (*ang. private cloud*);
- javni oblak (*ang. public cloud*);
- skupnostni oblak (*ang. community cloud*);
- hibridni oblak (*ang. hybrid cloud*).

#### Zasebni oblak

V kolikor je ponudnik virov hkrati tudi uporabnik virov, potem je nadzorni model zasebni oblak, kot je prikazano na Sliki 3.3. Uporabljajo ga predvsem podjetja oziroma organizacije za svoje namene, upravljanje in gostovanje pa lahko vodi bodisi podjetje, ki ga uporablja, ali pa ponudnik. Podjetja in

organizacije ga uporabljajo predvsem za izboljšanje poslovanja, zmanjševanje stroškov in dostopnost podatkov za zaposlene [12, 62].

### **Javni oblak**

V kolikor je ponudnik virov podjetje ali organizacije, ki omogočajo večstanovski (*ang. multi-tenant*) oblak in ponujajo uporabo virov drugim podjetjem ali organizacijam, ki nimajo potrebe in finančnih zmožnosti za vzpostavitev oblaka na lastni strojni opremi, potem gre za javni oblak, prikazano na Sliki 3.3. Glede na arhitekturo se ne razlikuje od zasebnega oblaka. Razlika je le v tem, da pri javnem oblaku podatkovni promet poteka po celotnem spletu (internetu), medtem ko pri zasebnem oblaku poteka zgolj v zasebnem omrežju v okviru podjetja [12, 62].

### **Skupnostni oblak**

V kolikor je ponudnik virov organizacija ali podjetje, ki svoje vire ponuja v najem določeni skupini preostalih organizacij, gre za nadzorni model skupnostnega oblaka, kot prikazuje Slika 3.3. Določena skupina organizacij predstavlja enake skupne interese in značilnosti, kot so dejavnost, varnostne zahteve, politika podjetja in druge. Oblak lahko upravlja in nadzoruje omenjena skupina podjetij ali pa ponudnik [12, 62].

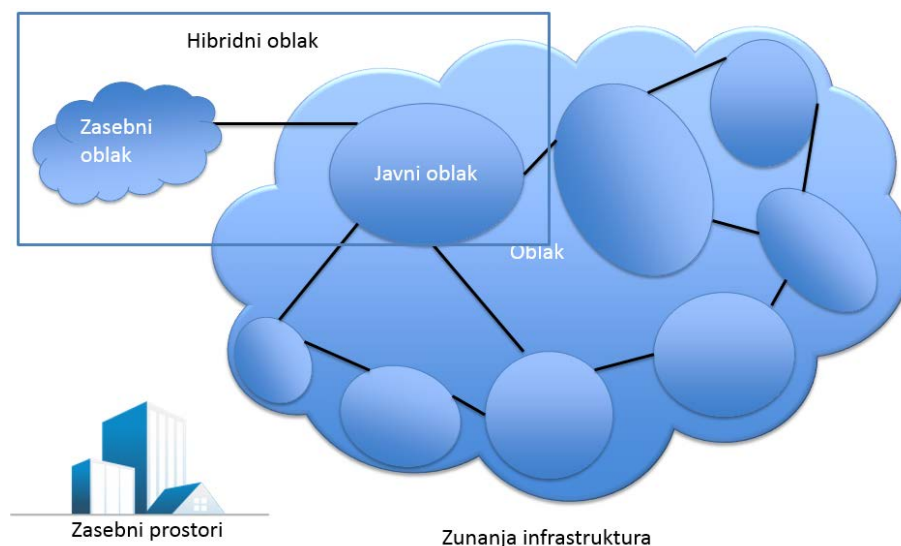
### **Hibridni oblak**

V kolikor uporabnik virov za svoje poslovne procese uporablja kombinacijo več nadzornih modelov oblaka, gre za hibridni oblak, ki je prikazan na Sliki 3.3. Uporabniki pogosto razdelijo svoje procese na zasebnem oblaku, manj občutljive procese oziroma procese s potrebami po povečljivosti pa v javnem oblaku. Vsi nadzorni modeli oblakov morajo uporabljati standardizirano in namensko tehnologijo, ki omogoča prenosljivost podatkov in aplikacij [12, 62].

### **Razširjanje v javni oblak**

V kolikor uporabnik uporablja enakovredno tako zasebne kot javne vire, vendar le v določenih časovnih obdobjih, ko zasebni oblak ne zagotavlja dovolj

virov, zasebni oblak svoje poslovne procese razporedi (*ang. load balancing*) med zasebnim in javnim oblakom (*ang. cloudbursting*), ki se ga uporablja kot »rezervno kapaciteto« zasebnemu. Na ta način oblak še vedno zagotavlja svojo povečljivost (skalabilnost) in performančne zmogljivosti. Obstaja tudi tako imenovano posredovanje do ponudnikov računalništva v oblaku (*ang. cloud brokerage* oziroma *cloud bursting*), ki je posebna oblika nadzornega modela. Posrednik (*ang. cloud broker*) glede na trenutne razmere in potrebe uporabniku storitve posreduje najprimernejšega ponudnika virov [12].



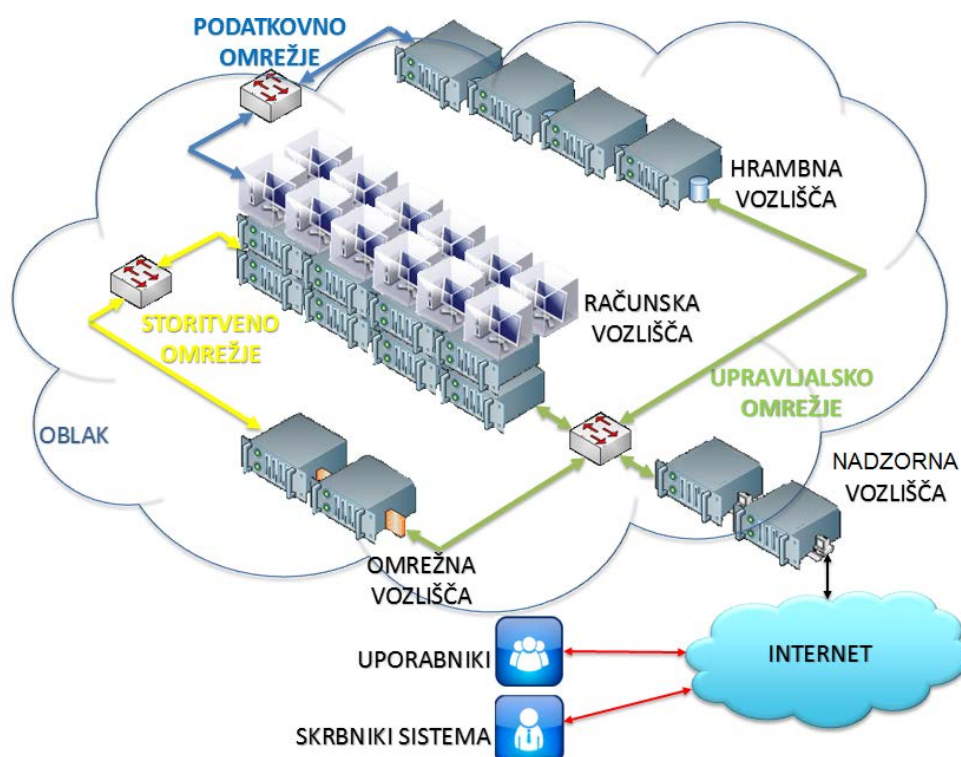
Slika 3.3: Postavitveni modeli računalništva v oblaku.

### 3.5 Splošna arhitektura infrastrukture

Platforma OpenStack uporablja infrastrukturo in jo omogoča kot storitev. Potrebno je pogledati, kaj tvori infrastrukturo. Njena arhitektura predstavlja fizično postavitev strežnikov oziroma vozlišč in povezave med njimi ter je prikazana na Sliki 3.4. Splošna arhitektura vsake infrastrukture vključuje vozlišča za hranjenje podatkov oziroma podatkovno skladišče (*ang. data*



*storage*), ki vključuje diskovna polja včasih tudi tehnologiji SAN in NAS [50] ter omogoča shranjevanje podatkov uporabnikov oziroma navideznih strojev, ki se izvajajo na računskih vozliščih (*ang. computing nodes*) in so opisani v naslednjem poglavju. Nadzorna vozlišča (*ang. controle node*) zagotavljajo pravilno delovanje celotnega sistema in omogočajo dostop preko spletnih vmesnikov ter CLI-jev končnim uporabnikom oziroma sistemskim skrbnikom, preko katere upravljajo oblak. Omrežna vozlišča omogočajo povezavo med navideznimi stroji in internetom. Na voljo je zunanje (*ang. external*) in notranje (*ang. internal network*) omrežje. Preko zunanjega omrežja se povezujejo uporabniki in skrbniki sistemov, preko notranjega pa služi samemu oblaku. Podatkovno omrežje je namenjeno prenosu podatkov med podatkovnim skladiščem in računskimi vozlišči, storitveno omogoča prenos podatkov ter operacij med računskimi vozlišči in internetom, upravljavsko zagotavlja upravljanje ter preverjanje pravilnosti delovanja celotnega oblaka [73]. Glede na namen uporabe oblaka se določi, koliko vozlišč bo upravljalo določeno funkcionalnost.



Slika 3.4: Splošna arhitektura storitvenega modela IaaS.

### 3.6 Prednosti in nevarnosti uporabe oblaka

Uporaba oblaka prinaša številne prednosti, vendar tudi nekaj pasti [46]. Prednosti so:

- **nižji stroški infrastrukture:** procesorsko moč oblaka lahko podjetja izkoristijo. To podjetjem omogoča prihranitev investicije za nadgrajevanje obstoječega sistema IT in dokupovanja programske opreme;
- **manj vzdrževanja strojne in programske opreme:** storitve oblaka pripomorejo k nižjim stroškom vzdrževanja tako strojne kot programske opreme, celo k zmanjševanju obeh oprem;

- **nižji stroški programske opreme:** stroški licenc programske opreme so lahko zelo visoki. Namesto nakupa in nameščanja programske opreme na računalnikih v podjetju je mogoče uporabiti programsko opremo, ki je nameščena na oblaku. Pri tem se ne plačujejo licence, temveč najem aplikacij, ki je bistveno cenejši;
- **redno izvajanje posodobljanj programske opreme:** v kolikor podjetje uporablja aplikacije kot storitev oblaka, potem prihrani pri nakupu posodobitev programske opreme in času za namestitve, saj spletno zasnovane aplikacije samodejno izvajajo posodobitve;
- **neskončna računska moč in neomejena shramba:** ko se podjetje poveže na oblak, svoje računske in prostorske zmogljivosti poveča; teoretično lahko uporablja vso infrastrukturo oblaka, kar je lahko nekaj tisoč strežnikov in ni več omejeno samo na lastne zmogljivosti;

Slabosti so:

- **zasebnost in lokacija podatkov:** podjetja nimajo informacije o tem, kje se podatki točno nahajajo in tudi niso povsem prepričana o zasebnosti podatkov, saj lahko teoretično do njih dostopa ponudnik;
- **varnost in izguba podatkov:** oblak je lahko neprimerno zaščiten ali pa je tarča vdorov in napadov. Tako lahko do podatkov dostopajo tretje osebe, podatki na oblaku se zapišejo na več strežnikov, še vedno pa lahko podjetje ostane brez podatkov, če ni izdelovalo varnostnih kopij in podobno;
- **nenehna in dovolj hitra internetna povezava:** za uporabo storitev oblaka je potrebno imeti nenehno ter hitro internetno povezavo, česar včasih ne more zagotoviti niti ponudnik internetnih storitev;
- **omejene zmožnosti:** določene aplikacije v oblaku ne podpirajo vseh funkcionalnosti, ki jih podpirajo drugače, nameščene na računalnikih;

- **počasno izvajanje storitev:** zaradi nenehnega prenašanja podatkov med oblakom in računalnikom se lahko določene storitve kljub hitri internetni povezavi počasneje izvajajo;

## Poglavje 4

# Virtualizacija kot sestavni del infrastrukture oblaka

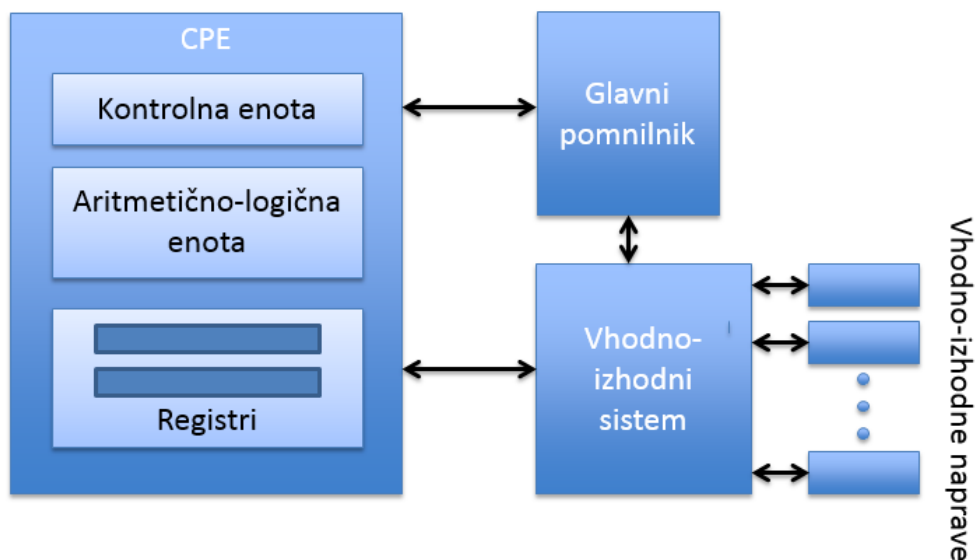
Virtualizacija predstavlja abstrakcijsko plast med programi in strojno opremo ter je skupna vsem virtualizacijskim tehnikam, ki jih bomo spoznali v tem poglavju. Namen uporabe virtualizacije je ključnega pomena za zapleteno infrastrukturo oblaka, saj povečuje izkoristek strojne opreme, zmanjša porabo električne energije, omogoča povečljivost in elastičnosti podatkovnih centrov, uporabnikom omogoča dostop do virov na zahtevo, poenostavi opravila skrbnika sistema in zagotoviti enostaven sistem za upravljanje ter nadzor nad strojno opremo in podpora več operacijskim sistemom. V poslovnem okolju se lahko uporabi tudi kot samostojna rešitev [40].

### 4.1 Resnični stroj

V matematiki so se skozi čas razvili različni modeli računanja, ki so formalno zapisane abstrakcije, ki zavzemajo vse, kar je bistveno za postopek reševanja problemov, porabljen čas in drugi vir. V računalništvu model računanja določa množico dopustnih operacij za reševanje teh problemov, računsko

zahtevnost in velikost pomnilnega prostora. Resnični stroj je računalnik, ki lahko rešljive probleme reši v nekem času po določenem modelu računanja in pri tem mora imeti vire, ki jih bomo spoznali v nadaljevanju. Večina današnjih računalnikov deluje na osnovi von Neummanove arhitekture. Kot je prikazano na Sliki 4.1, je Von Neummanov računalnik sestavljen iz treh enot, in sicer iz [42]

- **centralno procesne enote** (*ang. Central processing unit*, v nadaljevanju CPE) oziroma procesorja, ki je razdeljen na kontrolno, aritmetično-logično enoto in registre. Računske operacije se izvajajo v aritmetično-logični enoti. Register je majhen pomnilnik, v katerega procesor začasno shranjuje operande in ukaze. Operandi so podatki (številke). Kontrolna enota skrbi za prevzemanje ukazov in operandov ter za aktiviranje operacij nad aritmetično-logično enoto;
- **glavnega pomnilnika** (*ang. Random-access memory*, v nadaljevanju RAM) v katerem so shranjeni ukazi in operandi;
- **vhodno/izhodnega sistema** (*ang. Input/Output System*), ki je namenjen pretvorbi informacij med zunanjim svetom (človekom) in računalnikom. Vhodno/izhodni sistem so tipkovnice, monitorji, tiskalniki in pomožni pomnilnik za masivno ter trajno shranjevanje podatkov (*ang. Hard disk drive*, disk, v nadaljevanju HDD).



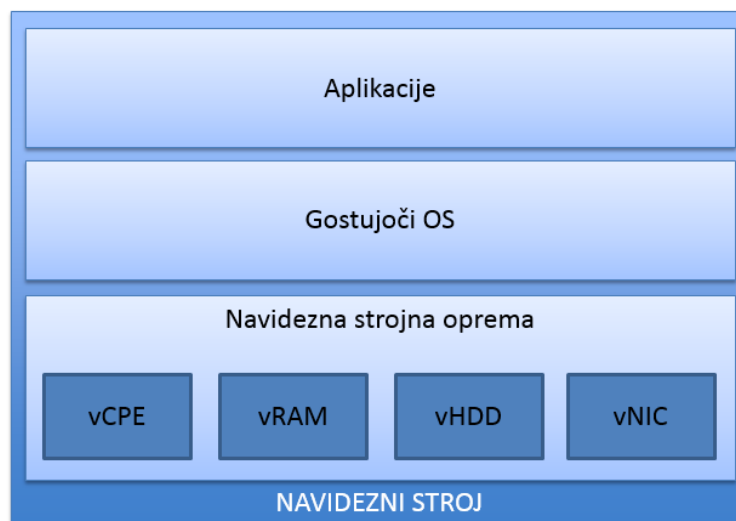
Slika 4.1: Arhitektura von Neummanovega računalnika.

Stroj za svoje delovanje potrebuje operacijski sistem in aplikacije, ki so nekakšen vmesnik med strojno opremo ter uporabnikom. Von Neummanov računalnik deluje tako, da bere (*ang. fetch*) ukaze iz pomnilnika in izvršuje te ukaze (*ang. execute*). Vsak ukaz vsebuje informacijo o operaciji in operandih, na katerih se bo operacija izvedla. Nato se ponovno prebere in izvede ukaz.

## 4.2 Navidezni stroj

Navidezni stroj (*ang. virtual machine*) posnema obnašanje resničnega stroja. Navidezni stroj je emulacija računalniškega sistema, deluje na osnovi računalniške arhitekture in funkcij resničnega ali hipotetičnega računalnika, nima lastne fizične strojne opreme, temveč za svoje delovanje uporablja navidezno strojno opremo oziroma vire, ki so enaki kot pri resničnem stroju: vCPE, vRAM, vHDD in mrežno kartico (*ang. Network interface controller, v nadaljevanju vNIC*), kot je prikazano na Sliki 4.2. Na njem je nameščen gostujoči operacijski sistem, ki glede na vrsto strežniške virtualizacijske tehnike

lahko loči med navideznim ali resničnem okoljem [40].



Slika 4.2: Navidezni računalnik za svoje delovanje uporablja navidezne vire.

Če želimo, da je navidezni stroj enakovreden resničnemu, mora imeti naslednje lastnosti [40]:

- **združljivost:** navidezni stroj podpira delovanje programske opreme tako kot resnični stroj;
- **strojno neodvisnost:** navidezni stroj je popolnoma neodvisen od fizične strojne opreme in ga je tako mogoče prenašati med fizičnimi računalniki;
- **osamitev:** navidezni stroji si med sabo delijo fizično opremo, vendar delujejo kot samostojni resnični stroji;
- **zakrivanje:** navidezni stroj vsebuje programski vsebnik (*ang. container*), kamor je enakpsulirana programska oprema; tako ga je mogoče enostavno prenašati med fizičnimi strežniki.



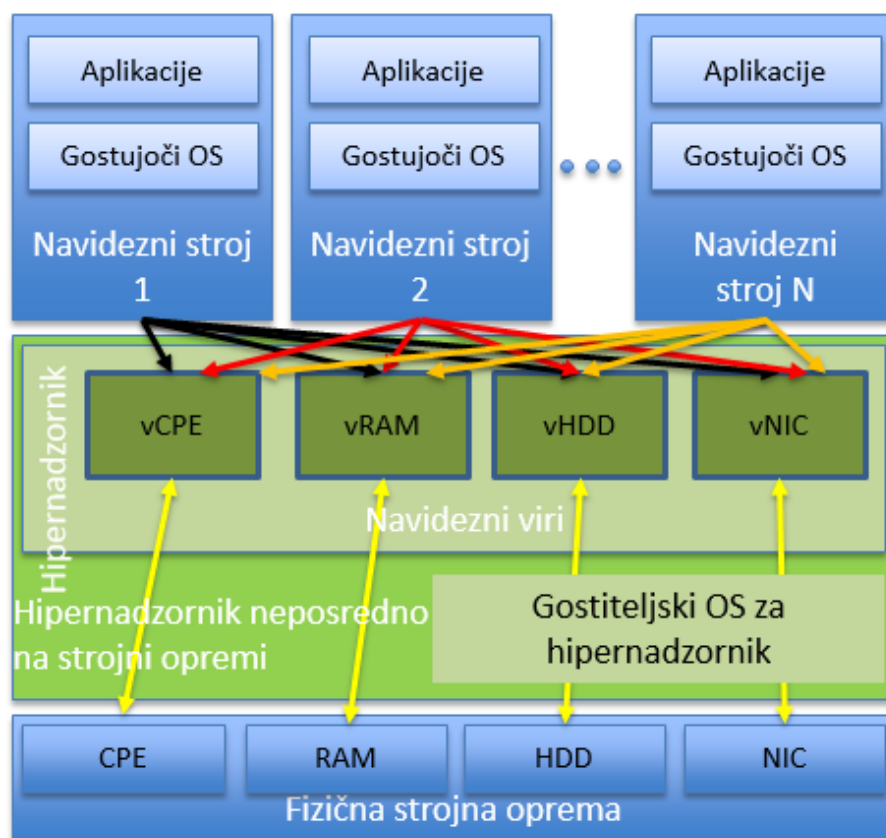
## 4.3 Razvoj virtualizacije

Pojav virtualizacije seže v šestdeseta leta prejšnjega stoletja. Vpeljalo jo je podjetje IBM v svoje osrednje računalnike z namenom vzporednega izvajanja več programov in aplikaciji na isti strojni opremi ter posledično večjega izkoristka strojne opreme, kar je tudi njen glavni cilj. Leta 1960 je bil superračunalnik Atalas eden izmed prvih računalnikov, ki so omogočali časovno deljenje (*ang. time sharing*) virov uporabnikom in multi programiranje (*ang. multiprogramming*), ki ustvari iluzijo vzporednega izvajanja aplikacij. Takrat se je prvič pojavil hipernadzornik (*ang. hypervisor*), na katerem se je izvajal del procesov operacijskega sistema, preostali del pa se je izvajal v komponenti za izvrševanje uporabniških programov. V nadaljnjih inačicah Atlasa so še dodali navidezni spomin in odstranjevanje pomnilnika (*ang. paging*). Tako so ločili dejanske naslove podatkov od logičnih naslovov in bistveno pridobili na hitrosti izvajanja. Sredi sedemdesetih let se je pojavil navidezni stroj (*ang. virtual machine*), ki je hkrati uporabljal tako strojno kot programsko opremo, v računalniškem sistemu IBM M44/44X, ki je bil zasnovan na Atalasuvi arhitekturi [74]. Nato so razvili še sistem IBM 7094, ki je omogočal združljivost deljenja časa (*ang. compatible time sharing sistem*), katerega glavna prednost je združljivost operacijskega sistema v ospredju z operacijskim sistemom, ki so se izvajali v ločenih navidezni strojih v ozadju in dostopali do večine vhodno-izhodnih naprav [14]. V tem času je zmogljivost procesorjev naraščala po Moorovem zakonu, ki pravi, da se na vsako leto in pol njihova zmogljivost podvoji [48]. Zaradi varnostnih razlogov, da bi pokvarjena aplikacija vplivala na drugo aplikacijo, ki se izvaja na enakem strežniku, so se aplikacije večinoma izvajale vsaka na svojem strežniku. Zasedenost strežnikov je bila zaradi tega v tem času med 10 in 15 odstotkov, kar je predstavljalo veliko vzdrževalnega dela in stroškov. Zato je podjetje VMware [81] v devetdesetih letih začelo z razvojem strežniške virtualizacije na arhitekturi x86, ki bi omogočala delitev infrastrukture, poenostavitev podatkovnih centrov in zmanjševanje operativnih stroškov, kar je pripomoglo k hitremu napredku virtualizacije do leta

2000.

## 4.4 Hipernadzornik

Hipernadzornik (*ang. hypervisor* ali *Virtual Machine Monitor*) je virtualizacijska platforma, programska oprema ali strojna oprema, ki omogoča istočasno delovanje več navideznih strojev. Hipernadzornik je po delovanju podoben operacijskemu sistemu in zagotavlja navideznim strojem dostop do virov, kot so CPE, RAM in HDD, kot je prikazano na Sliki 4.3. Nahaja se lahko neposredno nad strojno opremo (*ang. native hypervisor – bare metal*) ali na operacijskem sistemu (*ang. hosted hypervisor*), ki ima vlogo njegovega gostitelja. Vse sistemske klice gostujočega operacijskega sistema prestreže, ustrezno spremeni in posreduje gostiteljskemu operacijskemu sistemu. Izvaja osnovne operacije nad virtualnimi stroji, kot so ustvarjanje, zastavljanje in uničevanje. Prav tako omogoča osamitev in odpornost infrastrukture. Tako nestabilnost enega virtualnega stroja nima vpliva na delovanje ostalih strojev in v primeru fizične okvare strežnika virtualni stroj prenese na drugi fizični strežnik [40].



Slika 4.3: Umestitev hipernadzornika in njegovo delovanje.

## 4.5 Virtualizacija in virtualizacijske tehnike

Pogosto se pojavlja zamenjava virtualizacije z računalništvom v oblaku, vendar temu ni tako. Virtualizacija je sestavni del računalništva v oblaku, lahko se pa tudi samostojno uporabi kot rešitev nekega sistema IT. Definicija virtualizacije Ameriškega nacionalnega inštituta za standarde in tehnologijo se glasi [41]:

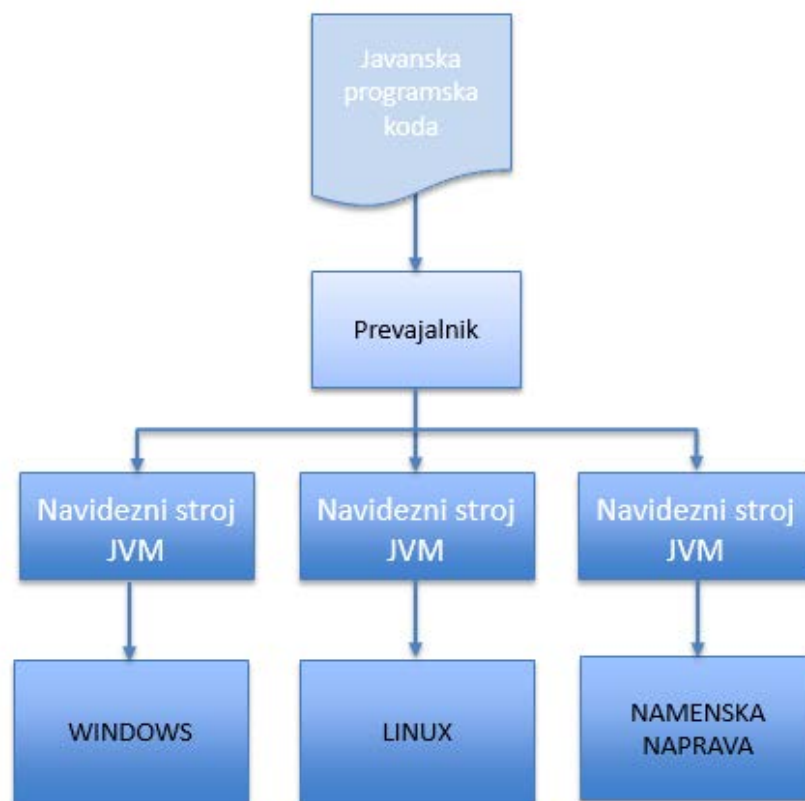
*Virtualizacija je simulacija programske in/ali strojne opreme, nekateri tudi rečejo druga programska oprema. To simulirano okolje se imenuje navidezni stroj.*

Virtualizacija uvede abstrkcijsko plast med fizičnimi in navideznimi okolji.

Združuje kratkotrajne vire, kot so CPE, spomin, hramba, omrežje, krmilnik hrambe in omrežja, vhodnih in izhodnih naprav v skupni bazen virov, iz katerega glede na posamične potrebe navideznih strojev jim dodeljuje njihove navidezne vire. Virtualizacija omogoča optimizacijsko tehniko izenačevanja obremenitve, ki je ključnega pomena za infrastrukturo oblaka, saj preprečuje njeno preobremenitev oziroma uravnoteženo izrabljaja vire, omogoča povečljivost in z redundantnostjo odpornost oblaka. V našem primeru je virtualizacija infrastrukture ključna in je prikazana v Dodatku B, saj omogoča migracijo navideznega stroja na ustrezen fizični strežnik, izdelovanje slike datotečnega sistema, na katerem je nameščen operacijski sistem in systemske nastavitve, gonilniki, nameščene aplikacije, podatki ter informacije o uporabniku. Nekateri formati slik so prenosljivi med procesorskimi arhitekturami in so neodvisni od vrste hipernadzornika [40].

Virtualizacija je zelo širok pojem. V tehničnem pogovoru virtualizacija pomeni virtualizacijo posameznih virov (CPE, pomnilnik, diski, mrežni viri) ali pa virtualizacijo strežnikov [41].

**Aplikacijska virtualizacija** (*ang. application virtualization*) je vmesnik med aplikacijo in strojno opremo. Vmesnik je potrebno namestiti na operacijski sistem. Je navidezna implementacija programskega vmesnika (API), na katerega se namesti aplikacija. Na ta način se aplikaciji omogoči okolje z vsemi potrebnimi viri, kot so registri, namestitvene datoteke in izvajalno okolje. Tako se aplikacija izolira, nima vpliva na ostale aplikacije, ne posega v operacijski sistem, ni je potrebno nameščati in spreminjati za prenašanje med različnimi operacijski sistemi. Takšne aplikacije se lahko izvajajo na enem primerku operacijskega sistema. Tipičen primer je javansko navidežno okolje (*ang. Java Virtual Machine – JVM*), ki omogoča poganjanje javanske programske kode na poljubnem operacijskem sistemu in je prikazano na Sliki 4.4 [41].



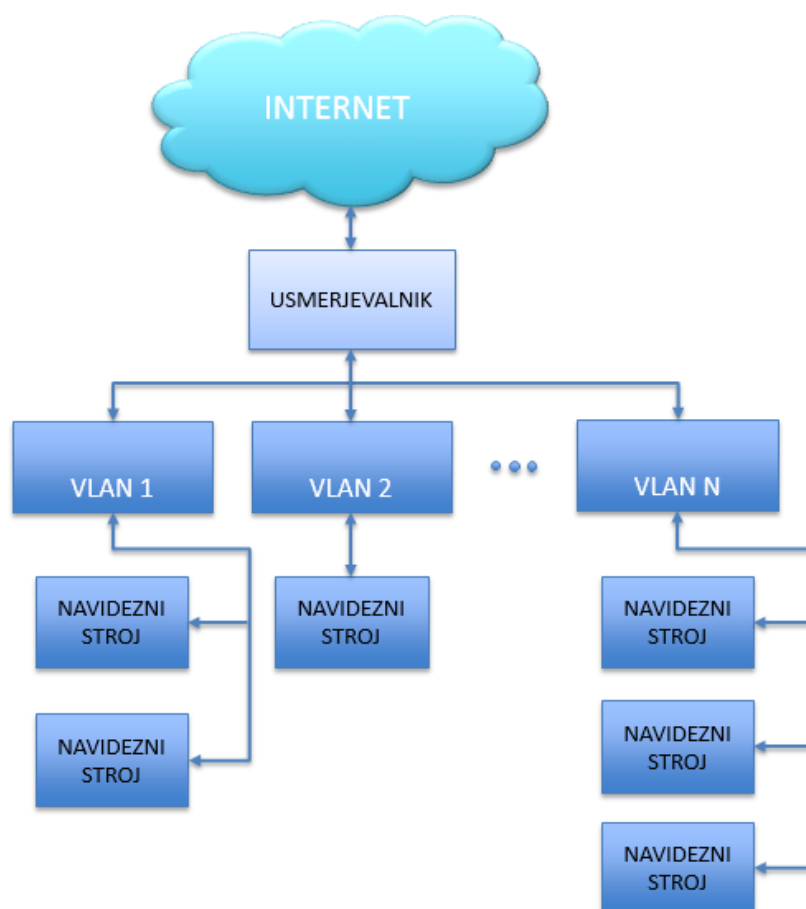
Slika 4.4: Aplikacijska virtualizacija programskega jezika Java.

### Virtualizacija strojne opreme

Za čim boljši izkoristek polne virtualizacije, je gostujočemu operacijskemu sistemu potrebno predstaviti navidezno strojno opremo, ki je čim bolj ekvivalentna fizični strojni opremi. Prav tako je potrebno ponuditi dodatne funkcije navidezne strojne opreme.

- **Virtualizacija omrežja** (*ang. network virtualization*): hipernadzornik lahko zagotovi navideznemu stroju navidezno omrežje za komunikacijo med navideznimi stroji in fizičnim omrežjem za dostop do interneta ter pri tem posnema fizično opremo omrežja, kot so usmerjevalniki, mostovi, stikala. Med njimi uporablja omrežni most (*ang. network bridge*), pri katerem ima gostujoči OS dostop do omrežne kartice gostiteljskega

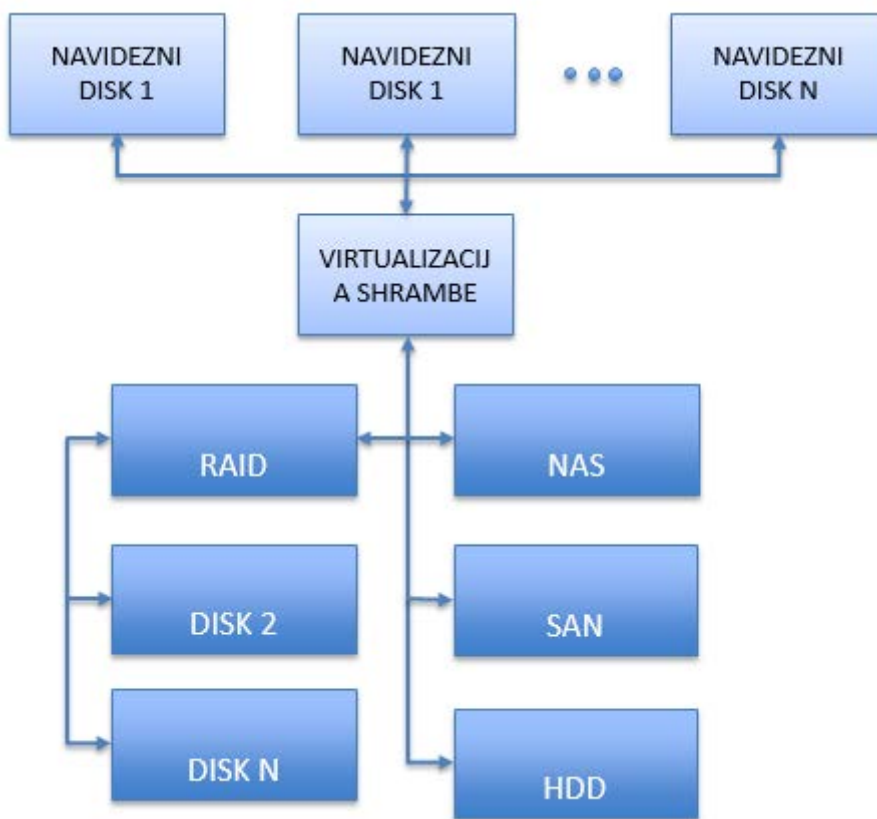
OS, pretvorbo samo naslovov IPv4 (NAT), ki jo izvaja hipernadzornik in prevaja mrežno kartico navideznega stroja z navideznim naslovom IP (*ang. Virtual IP*) v fizično omrežno kratico s fizičnim naslovom IP in gostiteljsko omrežje (*ang. host only network*), pri katerem se navideznim strojem doda navidezna mrežna kartica, preko katere se lahko navidezni stroji povezujejo ter imajo pri tem onemogočen dostop do fizičnega omrežja. Omogoča tudi ustvarjanje navideznih lokalnih omrežij (*ang. Virtual LAN – VLAN*) in navideznih zasebnih omrežij (*ang. Virtual private network – VPN*), kot je prikazan na Sliki 4.5. VLAN omogoča osamitev posameznega segmenta računalnikov, ki imajo na voljo svoje vire, kot so pasovna širina prenosa podatkov, spomin in procesor. VPN omogoča varno komunikacijo v javnih omrežjih, kot je internet, preko varnega komunikacijskega kanala za prenos občutljivih podatkov. Pogosto se uporablja za oddaljeni dostop zaposlenih do omrežja v podjetju [41].



Slika 4.5: Virtualizacija omrežja.

- **Virtualizacija hrambe:** (*ang. storage virtualization*) hipernadzornik lahko na več načinov simulira hrambo podatkov. Vsi hipernadzorniki omogočijo navidezni disk gostujočim OS-om, bolj napredni lahko ponudijo tudi dodatne možnosti shranjevanja, kot so povezovanje diskov v eno logično napravo (RAID) ali pa možnost shrambe podatkov naprav NAS in SAN, ki sta povezani v omrežje, kot je prikazano na Sliki 4.6. Navidezni disk je gostujočemu OS-u predstavljen kot slika diskovnega pogona, kar je v bistvu datoteka na gostiteljskem OS-u in se tudi z njo upravlja kot z navadno datoteko ter zaseže prostor na fizičnem disku. Gostujoči OS vse podatke piše in bere na/s slike diskovnega pogona.

Naprednejši hipernadzorniki gostujočim OS omogočajo hitrejši dostop do podatkov preko neposrednega dostopa do CD in HDD pogonov, elastične velikosti shrambe na zahtevo gostujočega OS-a, v ozadju skrbi za ustvarjanje varnostnih kopij podatkov, uporabnikom omogoča dostop do podatkov. Podatki so dosegljivi ne glede na lokacijo, čas ali vrsto naprave [41].



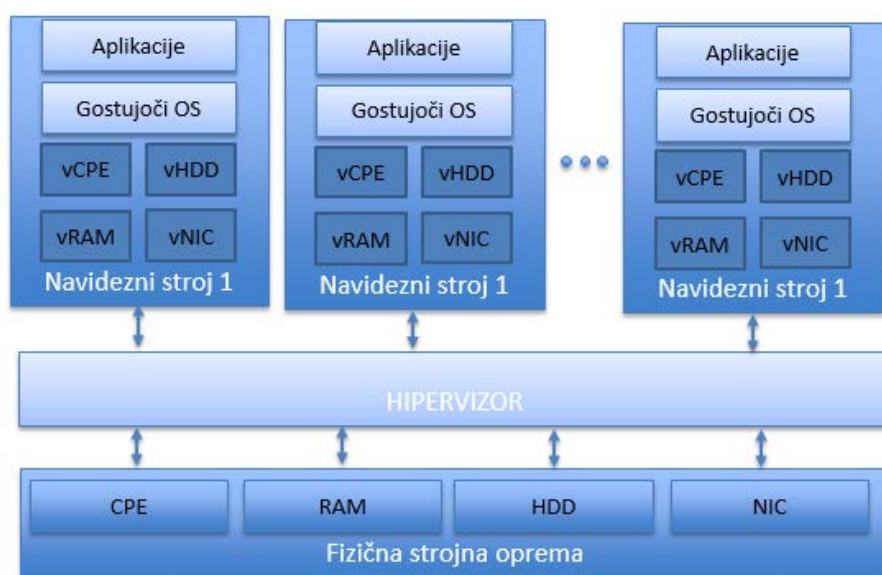
Slika 4.6: Virtualizacija hrambe.

Poznamo naslednje strežniške virtualizacije [41]:

- **emulacija strojne opreme** (*ang. hardware emulation*) je način virtualizacije, pri katerem hipernadzornik izvaja programsko emulacijo fizične strojne opreme, kot je prikazano na Sliki 4.7. Za vsak navidezni stroj ustvari navidezno strojno opremo, med samim izvajanjem pre-



streza njihove sistemske klice in upravlja dostop do fizičnih strojnih virov. Omenjeni način virtualizacije navideznemu stroju omogoča izvajanje gostujočega operacijskega sistema, ki je napisan za določeno drugo arhitekturo procesorja, na arhitekturi procesorja, ki ga uporablja fizični strežnik. Emulacija je najpočasnejša metoda virtualizacije, saj je potrebna pretvorba ukazov iz ene arhitekture v drugo [41];



Slika 4.7: Emulacija strojne opreme kot strežniška virtualizacija.

- **polna virtualizacija** (*ang. full virtualization*) izvaja polno simulacijo strojne opreme in tako navideznemu stroju omogoča, da uporablja velik nabor gostujočih operacijskih sistemov. Pri tem mora hipernadzornik pretvarjati ukaze navideznega stroja na strojne ukaze fizičnega procesorja, ker ne zahteva pretvorbo jedra gostujočega operacijskega sistema na arhitekturo fizičnega procesorja, temveč porabi veliko časa pri tem, kot je prikazano na Sliki 4.8. Določene uporabniške ukaze lahko izvaja neposredno na strojni opremi (modra poščica na sliki), kar pripomore k hitrejšemu delovanju, določene privilegirane ukaze pa ni mogoče virtualizirati in je potrebno binarno prevajanje (črna puščica na sliki),

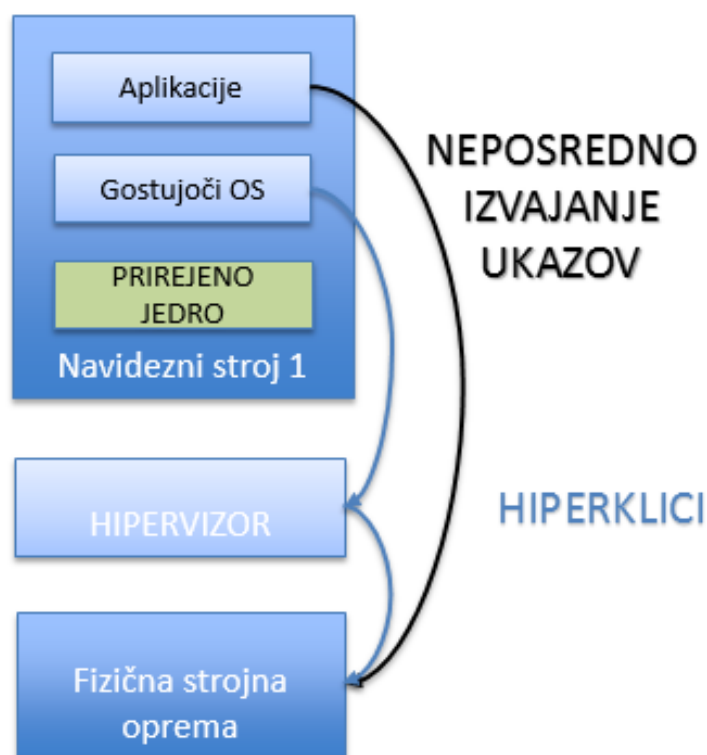
ki upočasni delovanje. Aplikacije se izvajajo na operacijskem sistemu, ki se nahaja na navideznem stroju, ki je gostujoči operacijski sistem. Hipernadzornik skrbi še za osamitev med virtualnimi stroji, dodelitev potrebnih navideznih virov in dostop do deljenih virov, kot so datoteke na gostiteljskem operacijskem sistemu. Pri tem se navidezni stroj ne zaveda, da je virtualiziran [41];



Slika 4.8: Polna virtualizacija kot strežniška virtualizacija.

- **paravirtualizacija** (*ang. paravirtualization*) izvaja delno simulacijo strojne opreme in pretvarjanja ukazov ter omogoča izvajanje določenih ukazov neposredno na strojni opremi fizičnega strežnika, kar predstavlja hitrejši dostop do diska in omrežja, kot je prikazano na Sliki 4.9. Zaradi tega ta način virtualizacije omogoča doseganje hitrosti delovanja, podobne nevvirtualiziranim okoljem. Zato uporablja posebne gonilnike,

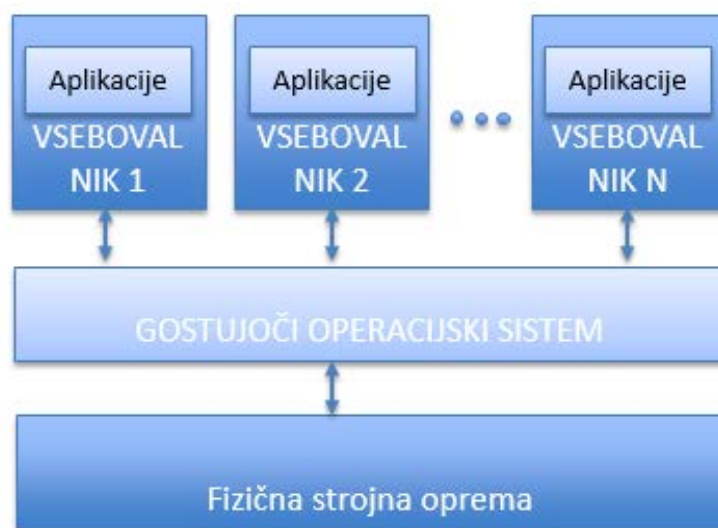
ki zamenjajo določene ukaze. Podobno kot prej, je uporabniške ukaze mogoče izvajati neposredno (črna puščica na sliki), pri izvajanju privilegiranih ukazov se uporabljajo hiperklici (*ang. hyper calls*, modra puščica). Hiperklici so ukazi, za katere gonilniki ne morejo poiskati ustrezne zamenjave. Zato je potrebno spremeniti jedro operacijskega sistema na navideznem stroju, na katerega namestimo prirejene gonilnike za navidezno strojno opremo. Različni hipernadzorniki ponujajo različne paravirtualizacije. Zaradi tega je slaba podpora nabora arhitektur gostujočega operacijskega sistema in je večinoma potrebno ustrezno prevesti operacijski sistem. Pri tem se navidezni stroj zaveda virtualizacije [41];



Slika 4.9: Paravirtualizacija kot strežniška virtualizacija.

- virtualizacija na nivoju operacijskega sistema (*ang. operating*

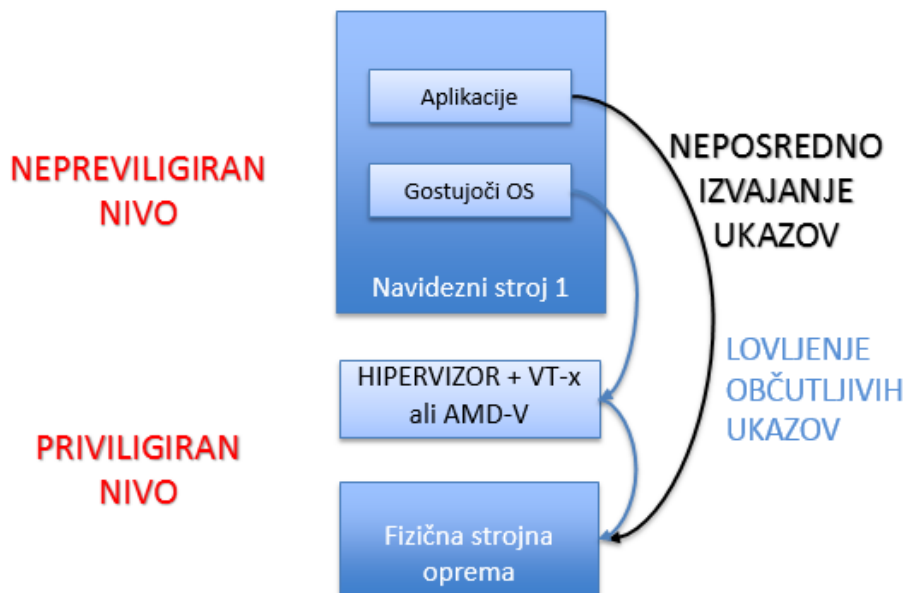
*system virtualization*) omogoča uporabo ene slike oz. skupno jedro gostiteljevega operacijskega sistema, na katerem se izvajajo med seboj neodvisni virtualizirani operacijski sistemi, ki izgledajo kot avtonomni sistemi (*ang. containers*) in ne kot navidezni stroji. Prikazani so na Sliki 4.10. Zaradi tega ni možnosti uporabe različnih operacijskih sistemov. Njegova prednost je predvsem v visoki hitrosti delovanja. Ta način virtualizacije ponujajo ponudniki spletnih storitev v obliki navideznih strežnikov [41];



Slika 4.10: Virtualizacija na nivoju operacijskega sistema.

- **strojno podprta virtualizacija** (*ang. hardware-assisted virtualization*) je podprta na strojnem nivoju fizičnega procesorja. Tako hipernadzorniku ni potrebno pretvarjati ukaze in navideznemu stroju dodeli čas fizičnega procesorja, kjer se strojno prevedejo ukazi, kar zagotavlja visoko hitrost delovanja virtualnega stroja. Ukazi se na navideznem stroju izvajajo v nepreviligranem načinu, kar omogoča neposredno izvajanje (črna puščica na sliki), so pa tudi previligrani ukazi (modra puščica na sliki), ki jih hipervizor ulovi in spremeni v primerne ukaze, ki jih procesor lahko izvede. Priviligrani ukazi so značilni za procesorko

arhitekturo x86, kot je prikazano na Sliki 4.11. Intelovi procesorji za strojno podporo virtualizacije ponujajo tehnologijo VT-x [31], AMD-jevi procesorji pa AMD-V [3, 41].



Slika 4.11: Strojno podprta virtualizacija.

## 4.6 Prednosti in slabosti virtualizacije

Virtualizacija ima predvsem dodano vrednost strežniške virtualizacije, ki je nepogrešljiva pri infrastrukturi oblaka in smo jih našli pri prednostih računalništva v oblaku v prejšnjem poglavju. Virtualizacija omogoča številne dodatne prednosti, kot so [40]:

- **konsolidacija vseh infrastrukturnih virov**, kot rešitev problema obvladovanja IT-ja v poslovnem okolju, kar predstavlja združevanje obstoječih sistemov v lažje obvladljivo in vzdržljivo okolje;
- **boljša izkoriščenost strojne opreme** je posledica delitve virov, ki jo izvaja virtualizacija. Tako se lahko več navideznih operacijskih sistemov in aplikacij integrira na enem fizičnem strežniku, ki deluje neodvisno drug od drugega;

- **neodvisnost strojne opreme** omogoča enostavno in hitro prenašanje navideznih strojev med različnimi okolji;
- **nadgrajevanje strojne opreme** brez potrebe po ponovnem nameščanju programske opreme;
- **večja varnost podatkov** je zagotovljena z enostavnim izdelovanjem varnostnih kopij, saj je potrebno le narediti posnetek navideznega diska, ki uporablja navidezni stroj.

Uvedba virtualiziranega okolja na drugi strani prinaša tudi slabosti, kot so [40]:

- **počasno izvajanje:** hitrost izvajanja navideznega okolja je nekoliko počasnejša, kar upočasni izvajanje navideznih strojev in aplikacij, zato se le določene aplikacije prenesejo v navidezno okolje;
- **strošek prehoda na navidezno okolje:** prehod predstavlja visok strošek v podjetjih, vendar se v večini primerov investicija dolgoročno odtehta;





## Poglavje 5

# Okolje OpenStack

V tem poglavju si bomo podrobneje pogledali platformo OpenStack, ki jo bomo v naslednjem poglavju tudi namestili. Zato je potrebno obravnavanje delovanja platforme in funkcionalnosti storitev.

### 5.1 Predstavitev OpenStacka

OpenStack je odprtokodna platforma, ki omogoča postavitev oblaka. Platforma je tudi projekt, ki nastaja na podlagi potreb velikih korporacij, ki ga dodelujejo svojim potrebam. Po storitvenem modelu oblaka infrastrukturo ponuja kot storitev, po načinu postavitve je lahko javen ali zaseben. Zadnja inačica platforme je Juno. V okviru tega diplomskega dela je predstavljena inačica Icehouse. Na vsakih šest mesecev izide nova inačica platforme. Smernice razvoja platforme potekajo v večji modularnosti posameznih komponent, ki se bodo lahko samostojno uporabljale v novejših storitvenih modelih računalništva v oblaku [65]. OpenStack je večinoma implementiran v programskem jeziku Python [64]. Zaradi številnih prednosti smo se odločili uporabiti omenjeno platformo [60]:

- omogoča različni načine postavljanja in vzdrževanje oblaka;

- zanesljivo delovanje in varnost podatkov;
- številne možnosti povečljivosti in prilagodljivosti;
- velik nabor hipernadzornikov in strojne opreme;
- uporaba spletne konzole namesto ukazne vrstice;
- velika odprtokodna skupnost in podpora s strani velikih računalniških podjetij;
- določene komponente je mogoče uporabiti kot samostojne komponente v integraciji z ostalimi rešitvami.

Tudi številna podjetja so prepoznala uporabo in se vključila v razvoj med več kot 350 podjetij so najbolj znani: Intel, HP, IBM in Cisco [78, 17].

Inačico Icehouse tvori deset komponent oziroma modulov. Za osnovno postavitev oblaka je potrebno namesti prve tri komponente OpenStacka. Vsaka komponenta predstavlja storitev, ki jo platforma uporablja za delovanje [9]:

- *Nova* - procesna (računska) storitev (*ang. compute service*);
- *Keystone* - identifikacijska storitev (*ang. identity service*);
- *Glance* - storitev za shranjevanje slik diskov navideznih strojev (*ang. image service*);
- *Neutron* - omrežna storitev (*ang. network service*);
- *Horizon* - spletni vmesnik oziroma nadzorna plošča platforme (*ang. dashboard*);
- *Cinder* - storitev za trajno shranjevanje podatkov v obliki blokovnih naprav (*ang. block storage service*);
- *Swift* - storitev, ki omogoča trajno shranjevanje podatkov v obliki objektov (*ang. object storage service*);
- *Heat* - orkestracijska storitev (*ang. orchestration service*);

- *Ceilometer* - modul za spremljanje in merjenje porabe virov (*ang. telemetry module*);
- *Trove* - ponuja povečljivo in zanesljivo storitev podatkovne baze (*ang. cloud database as a service*).

## 5.2 Komponente OpenStacka

Osnovne relacije med komponentami platforme so predstavljene na konceptualni arhitekturi v Dodatku C, v kateri se je tudi mogoče pogledati vse komponente, ki so predstavljene v nadaljevanju. Platformo je mogoče upravljati preko komponente oziroma preko klientov ukazne vrstice, ki omogoča pisanje ukazov ali pa poganjanje skript [9].

### 5.2.1 Računska storitev Nova

Računska storitev Nova je najbolj zapleten del platforme, ki ima v domeni upravljanje navideznih strojev, omrežij in določa dostop do oblaka. Za avtentikacijo uporablja storitve komponente Keystone. Za prvi zagon navideznega stroja uporablja sliko operacijskega sistema, ki jo hrani komponenta Glance. Za poganjanje navideznega stroja uporablja gonilnike za hipervizorje, ki smo jih razložili v četrtem poglavju. Tudi preko Amazonovega EC2 API-ja [28] je mogoče upravljati platformo, še prej pa je potrebno opraviti ustrezne nastavitve storitve. V Dodatku D so razloženi njeni procesi [22].

### 5.2.2 Identifikacijska storitev Keystone

Predstavlja nekakšno združitev platforme, saj združuje politike dostopov, kataloge storitev, stanovalce (*ang. tenants*), uporabnike in skrbnike platforme. Vsem storitvam in uporabnikom oblaka omogoča potrjevanje istovetnosti in

določa vrsto dostopa ter operacije, ki jih lahko izvaja. V katalogu storitev vzdržuje storitve, ki so na voljo, in hrani njihove dostopne točke, ki so naslov v omrežju, kjer lahko dostopamo do storitev platforme. Za hranjenje podatkov o avtentikaciji in avtorizaciji lahko uporablja relacijsko podatkovno bazo ali pare ključ – vrednost (*ang. pairs key – value*). V Dodatku D so predstavljeni vsi koncepti, ki jih ta storitev uporablja [36].

### 5.2.3 Storitev hrambe slik diskov Glance

Komponenta omogoča ustvarjanje zagonskih slik operacijskih sistemov, ki jih uporablja navidezni stroj, in shranjevanje slik diskov navideznih strojev v obliki posnetka (*ang. snapshot*), kjer so tudi podatki in aplikacije. Višje operacije (kot so: izdelava slik s posnetka navideznega stroja, pregled in brisanje slik, nastavljanje in brisanje njihovih metapodatkov) so v domeni komponente Nove. Metapodatki slik opisujejo parametre slik, kot so: CPE arhitektura, hipernadzornik, način delovanja navideznega stroja, samodejna nastavitve diska, podatke o gostujočem operacijskem sistemu slike. Metapodatki se shranjujejo v relacijsko podatkovno bazo. Slike navideznih strojev, ki so bile ustvarjene z Glancom, lahko shranjujemo v navaden datotečni sistem ali pa v obliki objektov, ki jih uporablja Swift [37]. V Dodatku D sta predstavljena API-ja komponente.

### 5.2.4 Omrežna storitev Neutron

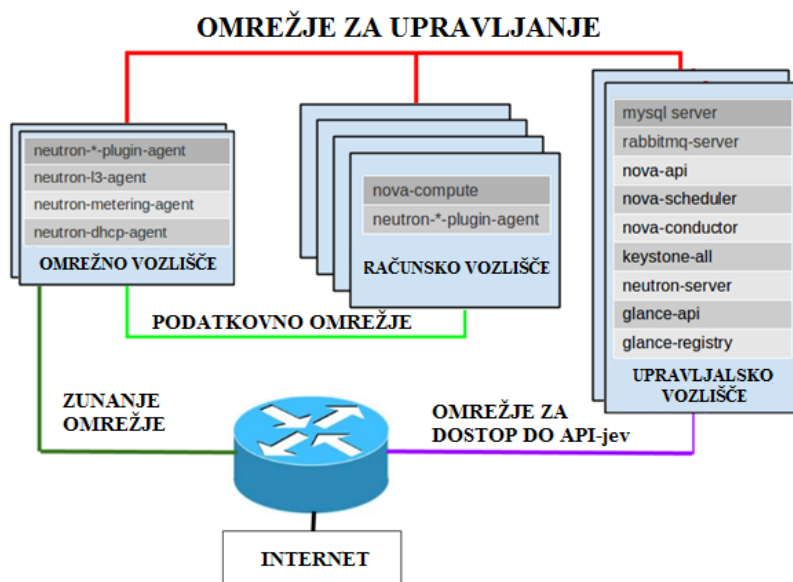
Komponenta Neutron zadošča vsem potrebam platforme po infrastrukturi, tako navideznega (*ang. Virtual Networking Infrastructure – VNI*) kot fizičnega omrežja (*ang. Physical Networking Infrastructure – PNI*) in omogoča dostop do navideznih strojev preko spleta. V zadnjih nekaj inačicah OpenStacka je mogoče uporabljati z omrežjem preko omenjene komponente ali s pomočjo storitve *nova-network*, ki je po funkcionalnosti omejena glede na namensko komponento. Komponente posnemajo lastnosti fizičnega omrežja, saj

omogočajo tvorjenje omrežij, podomrežij, usmerjanje prometa, napredne topologije navideznih omrežij, požarne zidove (*ang. firewalls*), izenačevanje obremenitve in zasebna navidezna omrežja. Ponuja več omrežij, in sicer:

- zunanje omrežje (*ang. external network*), preko katerega omogoča dostop do navideznih strojev iz zunanjega sveta in pri tem uporablja storitev NAT (*ang. network address translation*), ki privede naslov IP, ki ga uporablja javno omrežje, v drug naslov IP, ki ga prepozna omrežje platforme;
- notranje omrežje (*ang. internal network*), ki služi navideznim strojem;
- omrežje stanovalcev, ki omogoča dostop do navideznih strojev znotraj notranjega omrežja.

Za dostopanje do navideznih strojev iz zunanjega sveta je med omrežji potrebno imeti usmerjevalnike, ki imajo en prehod na zunanje omrežje (*ang. gateway*) in vmesnike do podomrežij, v katerih se nahajajo navidezni stroji. Plavajoči naslovi IP se dodelijo projektom in instancam znotraj projekta. Projekt je logična povezava med uporabniki in računsko storitvijo, ki definira dostop do slik navideznih strojev. Varnostne skupine določajo, kakšen promet lahko poteka proti instanci. V oblaku je več omrežij, kot je prikazano na Sliki 5.1 [54]. Le-te so:

- omrežje za upravljanje (*ang. management network*), ki je namenjeno komunikaciji med komponentami platforme;
- podatkovno omrežje (*ang. data network*), namenjeno komunikaciji med navideznimi stroji, ki tečejo na več računskih vozliščih;
- omrežje za dostop do API-jev (*ang. API network*) omogoča zunanji dostop do navideznega omrežja;
- notranje omrežje (ki je že opisano zgoraj).



Slika 5.1: Različna omrežja znotraj oblaka OpenStack.

V Dodatku D so predstavljeni procesi, ki jih uporablja Neutron [53].

### 5.2.5 Storitev za shranjevanje blokovnih naprav Cinder

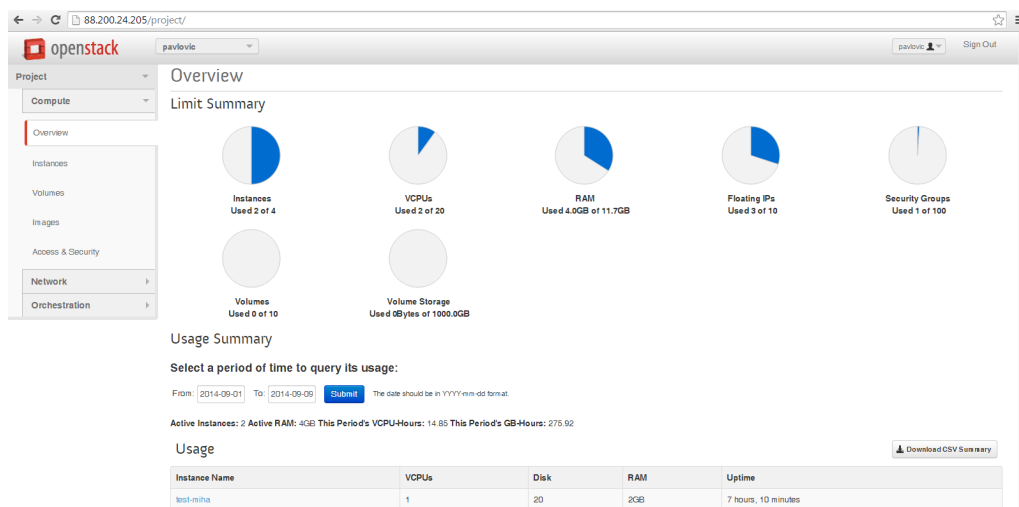
Komponenta Cinder omogoča obstojno hrambo podatkov navideznih strojev v obliki blokovne naprave (*ang. volume*), ki jo je potrebno pripeti (*ang. mount*) na navidezni stroj, nastaviti datotečni sistem in zapisovanje na blokovno napravo namesto na začasni disk. V osnovi navidezni stroj uporablja navidezni disk, ki je začasen (*ang. ephemeral*) in je na voljo dokler navidezni stroj obstaja. Če se stroj uniči, potem se vsebina navideznega diska izbriše. Če se navidezni stroj uniči, ob ponovnem tvorjenju blokovno napravo samo priklopimo in nadaljujemo z delom. Blokovno napravo lahko uporablja samo ena instanca naenkrat. Narediti je mogoče tudi varnostno kopijo blokovne naprave in jo shraniti kot objekt v komponento Swift. V Dodatku D je pregled API-jev, ki jih uporablja ta komponenta [13].

### 5.2.6 Storitev za shranjevanje objektov Swift

Komponenta Swift omogoča redundantno, visoko zmogljivo in povečljivo hrambo tako nestrukturiranih podatkov v obliki objektov kot slik, ki jih omogoča komponenta Glance. Lahko se jo namesti kot samostojno komponento na gručo strežnikov in tako podjetju ponuja prostor za trajno shrambo objektov. Nima neposredne vloge datotečnega strežnika. Objekt predstavlja osnovno entiteto za shranjevanje statičnih podatkov in njihovih metapodatkov, ki predstavljajo pare ključ – vrednost in hranijo: velikost podatkov, naslov IP gostitelja, ID gostitelja. Shranjeni podatki so nekompresirani in nešifrirani. Vsak objekt je repliciran na več strežnikov. V primeru odpovedi enega strežnika se objekt nahaja na drugem strežniku. Skrbniki sistema in uporabniki se morajo najprej avtentificirati preko Keystone, ki poteka skozi srednji sloj vmesnika WSGI. Operacije Swifta se izvajajo preko spletne storitve REST API (API z zahtevami HTTP). Za shranjevanje uporabniških računov Swift uporablja relacijsko podatkovno bazo [55]. V Dodatku D so predstavljeni procesi, ki omogočajo delovanje Swifta [59].

### 5.2.7 Spletni vmesnik Horizon

Uporabniki in skrbniki sistemov lahko preko vmesnika CLI ali preko spletnega vmesnika Horizon upravljajo z vsemi storitvami oblaka in viri platforme. Horizon je pregledna spletna konzola, ki ponuja večino zmogljivosti ukazne vrstice. Uporablja lastno podatkovno bazo, v katero shranjuje prijave, podatke o ustvarjenih instancah, izkoriščenosti navideznih strojev in druge. Spletni vmesnik lahko dostopa samo do javnih API storitev [25]. Na Sliki 5.2 je prikazan spletni vmesnik.



Slika 5.2: Spletni vmesnik za upravljanje OpenStacka.

### 5.2.8 Storitev za izvajanje opazovanja Ceilometer

Komponenta Ceilometer opazuje celotni sistem in zbira podatke o uporabi infrastrukture. Podpira več hipernadzornikov, od katerih dobiva podatke o virih, ki jih uporabljajo navidezni stroji. Tako ponudnikom oblčnih storitev omogoča potrebne podatke za kakovostno in upravičeno zaračunavanje svojih storitev. Do storitve je mogoče dostopati preko REST API-ja, CLI in spletnega vmesnika. Meritve shranjuje tako v relacijske kot dokumentne podatkovne baze [8]. V Dodatku D so predstavljeni agenti in procesi, ki so na voljo Ceilometeru.

### 5.2.9 Storitev orkestracije Heat

Komponenta Heat omogoča orkestracijo aplikacij in s pomočjo predlog infrastrukture pridobi potrebne vire, kot so navidezni stroji, plavajoči naslov IP, shramba slik diskov, varnostne skupine in uporabnike. Orkestracija izvaja razporeditev, usklajevanje in vodenje zapletenih računalniških sistemov, vmesni sloj, ki se nahaja med uporabniškimi programi in strojno opremo, ter storitev.



Torej gre za nekako predlogo za hitro tvorjenje potrebne infrastrukture [61]. Procesi komponente so v Dodatku D.

#### 5.2.10 Podatkovna baza kot storitev Trove

Komponenta Trove omogoča DaaS (*ang. Database as a Service*) znotraj platforme OpenStack. Komponenta uporabnikom omogoča uporabo podatkovne baze kot storitve. Podpira tako relacijske kot dokumentne podatkovne baze brez potrebe po predhodnem opravljanju opravil skrbnika sistema. Storitve zagotavlja osamitev virov in avtomatizira procese ter postopke, kot so: opravljanje nastavitve, spreminjanje baze, izdelava varnostnih kopij baze in opazovanja podatkovnih baz [26]. V Dodatku D si je mogoče pogledati procese, ki jih komponenta uporablja.

### 5.3 Logična arhitektura platforme OpenStack

Logična arhitektura podaja natančen pregled API-jev posameznih komponent, preko katerega se je mogoče povezati in uporabljati storitve ostalih komponent platforme. Tako so komponente sinhronizirane in oblak deluje kot celota. Komponente platforme delujejo po enem izmed spodaj opisanih načinov, in sicer [43]:

- prikriti proces (*ang. daemon*) teče v ozadju storitve in sprejema zahteve ter se nanje odziva;
- skripte (*ang. script*) se uporabljajo za namestitev okolja platforme in za zaganjanje testov za preverjanje delovanja komponent;
- CLI (*ang. command-line interface*) uporabnikom in skrbnikom sistema omogoča pošiljanje zahteve preko vmesnika API do storitev, ki jih uporablja in ponuja platforma.

V Dodatku E je podana shema logične arhitekture platforme, na kateri si je mogoče ogledati relacije med API-ji in komponentami. Zaradi visoke prilagodljivosti platforme je lahko končna logična arhitektura precej drugačna in v veliki meri odvisna od v naprej določene uporabe oblaka oziroma spreminjajočih se potreb podjetja. Podroben opis komponent in API-jev je podan v naslednjem poglavju.

## Poglavje 6

# Tečaj postavljanja okolja OpenStack

V uvodnem delu smo se spoznali s teoretičnim delom računalništva v oblaku, z navideznim strojem, virtualizacijo in s platformo OpenStack. V praktičnem delu se bomo predstavili s postopkom postavitve platforme. Kot smo že omenili, se tečaj v celoti izvaja znotraj e-Učilnice Moodle na spletnem naslovu <http://rtk2014.fri.uni-lj.si/>, v kateri se nahajajo vsa potrebna učna gradiva in navodila za dostop do OS Ubuntu Linux.

### 6.1 Uvod v tečaj

Celoten tečaj je razdeljen na štiri praktične učne enote. Vsaka enota je sestavljena iz osnovnega opisa učne enote, potrebnih predznanj učenca, opisa pridobljenih znanj oziroma ciljev naloge, ki jih je dobil učenec med opravljanjem naloge, podrobnega opisa postopka naloge ter iztočnice za možnosti nadaljnjega dela v okviru naloge.

Tečaj je razdeljen na štiri naloge, v katerih se bomo naučili:

- 1. naloga: preverili bomo sistemske zahteve platforme in predstavili osnovo delo z OS-om Ubuntu Linux, na katerega bomo namestili platformo. Storitve platforme za shranjevanje različnih podatkov uporabljajo podatkovno bazo in za medsebojno komunikacijo sporočilni strežnik;
- 2. naloga: namestili in nastavili bomo osnovne storitve platforme OpenStack: Keystone, Nova, Glance in Cinder;
- 3. naloga: ustvarili bomo uporabnike, stanovalce in storitve, ki jih bomo registrirali in ustvarili njihove dostopne točke;
- 4. naloga: ustvarili bomo vse potrebne vire za zagon navideznih strojev, torej navidezno omrežje, varnostne skupine, zagonsko sliko operacijskega sistema, predlogo navidezne strojne opreme in na koncu bomo še pognali navidezni stroj.

Preden začnemo, je potrebna še razlaga poimenovanj. Tekom tečaja se večkrat uporablja beseda *strežnik*, ki predstavlja naš fizični računalnik z nameščenim operacijskim sistemom, ki ga bomo predstavili v prvi nalogi. Pojavlja se tudi beseda *navidezni stroj* in ima enak pomen kot *instanca*.

Med tečajem je potrebno paziti na gesla in uporabnike. Uporabili bomo svoja gesla, vendar moramo pri tem paziti. Za lažje razumevanje in razlikovanje uporabe gesel med storitvami ter uporabniki smo gesla pisali kot *rabbitpass*, *adminpassword*, *keystonePass*, *mysqlpass* in *userpassword*. Za lažje razlikovanje med uporabniki in storitvami smo pisali *user*, *admin* in *keystoneUser*. IP naslov našega strežnika je 192.168.200.26; tega ne bomo uporabljali, ker imamo svoje naslove IP. V tečaju bomo navajali <ime.našega\_strežnika>, kar predstavlja ime našega strežnika. V tečaju se velikokrat ponovijo entitete identifikacije in avtorizacije, pri tem bomo dobili svoje entitete. Prav tako svoje ID-je (*ang. identifier*), ki predstavljajo enolično identifikacijo oznako ali številko uporabnika, stanovalca, storitev ali omrežja. Za lažje razumevanje določenih ukazov so podani tudi izpisi, zato bodo pri nam izpisi ustrezno

različni.

Postopek je povzet po dokumentaciji uporabe terminala v OS-u Ubuntu Linux [77] in namestitvi platforme OpenStack [39] ter uporabe dokumentacije o administraciji [1] in konfiguraciji [24].

## 6.2 Naloga 1: Preverjanje sistemskih zahtev OpenStacka in uvod v Ubuntu Linux

### Opis praktične učne enote:

V nalogi se bomo spoznali z osnovnimi ukazi navadnega in korenskega uporabnika v operacijskem sistemu Ubuntu Linux. Naučili se bomo, kakšne so sistemske zahteve platforme in kako jih lahko preverimo. Še posebej je pomembno, da preverimo strojno virtualizacijo. Namestili in nastavili bomo podatkovno bazo ter strežnik za sporočilne vrste, ki jih komponente potrebujejo za izvajanje procesov.

### Potrebna predznanja:

Učenec mora znati uporabljati računalnik, namen uporabe podatkovnih baz in sporočilnih vrst ter gradnikov računalniških sistemov.

### Pridobljena znanja:

Cilj naloge je, da učenec pridobi razumevanje sistemskih zahtev platforme in njihovo preverjanja na strežniku Ubuntu. Nauči se osnovne uporabe operacijskega sistema Ubuntu Linux, namestitve podatkovne in sporočilno baze.

### Postopek:

Postopek naloge je podan v poglavju 5 na e-tečaju.

### Za naprednejše udeležence tečaja:

Namesto podatkovne baze MySQL lahko poskusite z namestitvijo podatkovne baze SQLite, ali PostgreSQL. Za bazo sporočilnih vrst lahko poskusite namestiti Qpid [65]. Postopke lahko poiščete v dodatnih gradivih namestitev platforme.

## 6.3 Naloga 2: Namestitev in nastavitve komponent OpenStack

### Opis praktične učne enote:

V nalogo bomo namestili in nastavili komponente Keystone za identifikacijo ter avtorizacijo, Glance za shranjevanje slik diskov, Nova za zaganjanje navideznih strojev in Cider za pripenjanje blokovnih naprav. Tekom vaje bomo sproti preverjali, če smo pravilno nastavili nastavitve komponente.

### Potrebna predznanja:

Učenec mora poznati osnovno uporabo terminala v OS-u Ubuntu Linux, gradnikov računalniških sistemov in teoretični del platforme OpenStack.

### Pridobljena znanja:

Cilj naloge je učenca seznaniti s postopkom namestitev komponent in pridobiti razumevanje njihovih nastavitvev ter preverjanje njihovega delovanja po spreminjanju nastavitvev.

### Postopek:

Postopek naloge je podan v poglavju 6 na e-tečaju.

### Za naprednejše udeležence tečaja:

Platformo lahko poskusite namestiti na več fizičnih strežnikov, kot se platforma uporablja v produkcijskih okoljih. Glede na potrebe določite, koliko je potrebno nadzornih, računskih in podatkovnih fizičnih strežnikov. Prav

tako lahko poskusite z namestitvijo preostalih komponent. Postopke za obe opravili lahko poiščete v dodatnih gradivih namestitev platforme.

## 6.4 Naloga 3: Vzpostavitev okolja Open-Stack

### Opis praktične učne enote:

Pred začetkom uporabe platforme bomo vzpostavili potrebne entitete, kot so uporabnik, stanovalec in storitev. Ustvarili bomo uporabnike in jih dodelili stanovalcem. Tako bomo uporabnikom določili avtorizacijo. Tvorili bomo tudi storitve, jih določili stanovalcem, nato bomo registrirali in nastavili dostopne točke, na katerih se nahajajo API-ji storitve.

### Potrebna predznanja:

Učenec mora poznati osnovno uporabo terminala v OS-u Ubuntu Linux, gradnikov računalniških sistemov in teoretični del platforme OpenStack.

### Pridobljena znanja:

Cilj naloge je učenca seznaniti z opravili skrbnika sistema, ki so potrebni pred operativno uporabo platforme.

### Postopek:

Postopek naloge je podan v poglavju 7 na e-tečaju.

### Za naprednejše udeležence tečaja:

Ustvarite še druge stanovalce, določite jim dostopne pravice. Če ste v prejšnji nalogi opravili dodatno nalogo, potem lahko še preostale komponente registrirate in tovorite njihove dostopne točke. V dodatnih gradivih se lahko seznanite z vsemi možnostmi nastavljanja dostopnih pravic, ki se v produkcijskih okoljih izkažejo za nujno potrebne.

## 6.5    Naloga 4: Zagon navideznega stroja

### Opis praktične učne enote:

V nalogi se bomo pred zagonom navideznega stroja seznanili z vsemi potrebnimi opravili. Najprej bomo ustvarili omrežje, v katerega bomo vključili stroj; omrežju je potrebno določiti varnostno skupino, ki določa način dostopa do stroja. Za delovanje stroja, tako kot pri navadnem računalniku, je potrebno namestiti operacijski sistem, v našem primeru Linux Ubuntu, ki ga bomo prenesli z javno dostopnega spletnega skladišča in ga nastavili ob zagonu stroja. Pred zagonom je stroju potrebno dodeliti predlogo navidezne strojne opreme. Za dostop in uporabo stroja bomo nastavili še SSH ključ.

### Potrebna predznanja:

Učenec mora poznati uporabo terminala OS Ubuntu Linux, poznavanje delovanja platforme in navideznega stroja.

### Pridobljena znanja:

Cilj naloge je učenca seznaniti z ustvarjanjem vseh potrebnih navideznih virov in njihovo vlogo pri izvajanju navideznega stroja.

### Postopek:

Postopek naloge je podan v poglavju 8 na e-tečaju.

### Za naprednejše udeležence tečaja:

V dodatnem gradivu lahko preverite ukaze za upravljanje in prehode med različnimi stanji navideznega stroja.



## Poglavje 7

# Evalvacija tečaja

Namen empiričnega dela diplomskega dela je bil raziskati uporabnost in učinkovitost tečaja. Zanimalo nas je, če je tečaj pravilno oblikovno in vsebinsko zastavljen ter kakšne so možnosti izboljšave. Zato je bilo potrebo izvesti raziskavo na skupini učencev in pripraviti instrument za vrednotenja tečaja ter znanj, ki so ga učenci pridobili.

### 7.1 Cilj raziskave

Temeljni cilj raziskave je preveriti uporabnost in učinkovitost tečaja za postavljanje platforme OpenStack pri različnih uporabnikih. Želimo tudi ugotoviti morebitne pomanjkljivosti tečaja oziroma dobiti konkretne predloge izboljšav in jih posredovati pri nadaljnjem samostojnem učenju oziroma ugotovitve prenesti na posameznike, ki se bodo ukvarjali z izdelavo podobnih tečajev. Z namenom dobre vsebinske in oblikovne izvedbe tečaja smo v poglavju *Učenje v e-izobraževanju* poiskali smernice o izvedbi tečaja. Z namenom preverjanja kakovosti in obsežnosti tečaja smo si zastavili naslednja raziskovalna vprašanja:

- ali je tečaj celostno, vsebinsko in oblikovno primerno zastavljen?

- ali so posamezne učne enote kakovostno obravnavane in zajemajo širok spekter specifičnosti učencev, ki vključuje različna predznanja in zanimanja, pričakovanja, učni stil, motivacijo ter sposobnost samostojnega učenja?
- ali se pridobljeno znanje iz teoretičnih učnih enot lahko uporabi v praktičnih učnih enotah?
- ali so tematike v teoretičnih učnih enotah razumljive, podajajo dovolj potrebnih informacij in obrazložitev?
- ali so navodila in postopki v praktičnih učnih enotah razumljivi?
- katere učne cilje je učenec dosegel pri učnih enotah?

## 7.2 Metoda raziskovanja in raziskovalni vzorec

Uporabili smo deskriptivno, kavzalno-neeksperimentalno, empirično, kvalitativno metodo raziskovanja. Empirično raziskavo smo opravili na skupini desetih učencev iz konkretne populacije. Njihove značilnosti so prikazane v Tabeli 7.1.

### Razlaga stolpcev:

#1 – zaporedna številka učenca

#2 – spol

#3 – starost

#4 – status (Š je študent/ka, R – zaposlen kot računalničar, Z – zaposlen izven računalniške stroke)

#5 – ocena lastnega poznavanja arhitekture računalniških sistemov

#6 – ocena lastnega poznavanja informacijsko-telekomunikacijskih sistemov

#7 – ocena lastnega poznavanja računalništva v oblaku

#8 – ocena lastnega poznavanja virtualizacije

#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
1	M	22	Š	2	2	1	2	1	NE	nobeno	nobeno
2	M	29	R	3	4	4	4	3	DA	nobeno	nobeno
3	Ž	23	Š	1	2	1	1	1	NE	nobeno	nobeno
4	M	27	Š	4	3	4	3	4	DA	nobeno	nobeno
5	M	25	R	4	3	4	3	3	DA	nobeno	nobeno
6	M	25	R	4	4	3	3	4	DA	nobeno	nobeno
7	M	23	Z	2	2	1	1	2	NE	nobeno	nobeno
8	M	22	Š	3	2	2	1	2	NE	nobeno	nobeno
9	M	21	Z	2	1	1	1	1	DA	nobeno	nobeno
10	M	25	R	5	3	4	3	5	DA	nobeno	nobeno

Tabela 7.1: Značilnost vzorca desetih učencev.

#9 – ocena lastnega poznavanja uporabe terminala v Ubuntu Linux

#10 – uporaba orodij za virtualizacijo (Virtual Box, VMWare in drugi)

#11 – uporaba ene izmed platform za infrastrukturo oblaka

#12 – namestitev platforme za infrastrukturo oblaka

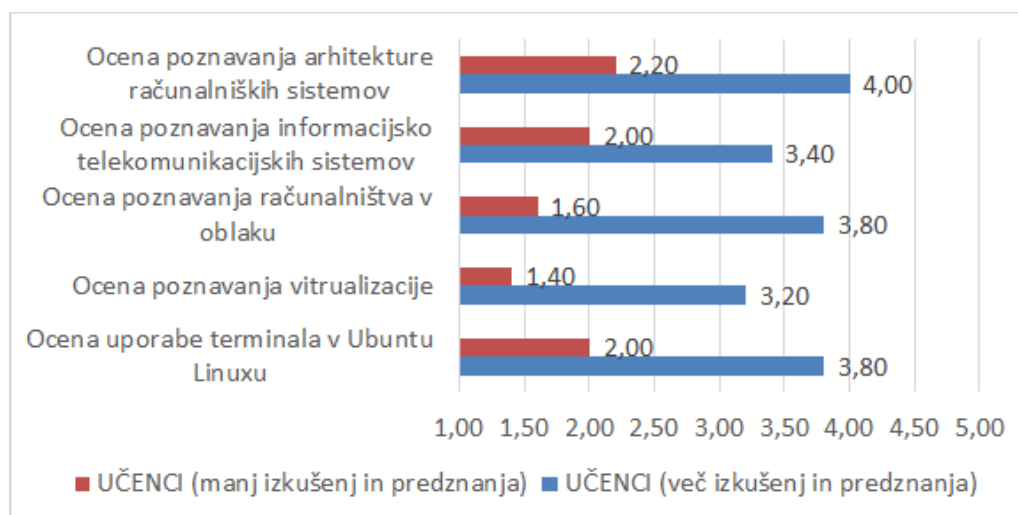
V vzorcu je 9 oseb moškega spola (90 %) in 1 oseba ženskega spola (10 %). Razpon starosti učencev je med 21 in 29 let, povprečna starost je 24,2 leti. Glede na status so štirje študenti (40 %), prav tako so štirje zaposleni kot računalničarji (40 %) in dva sta zaposlena izven računalniške stroke (20 %). En učenec (10 %) je ocenil svoje znanje poznavanja arhitekture računalniških sistemov z oceno 1, trije (30 %) z dve, trije (30 %) s prav dobro oceno in eden (10 %) z odlično, povprečna skupna ocena je točno 3. Poznavanje informacijskih telekomunikacijskih sistemov je eden (10 %) ocenil, da nič ne zna, največ učencev, ki so ocenili znanje z zadostno oceno, so štirje (40 %), trije (30 %) so prepričani v dobro oceno, dva (20 %) sta ocenila, da ga zelo dobro poznata, povprečna ocena vzorca je 2,6. Poznavanje računalništva so štirje ocenili (40 %), da ga sploh ne poznajo, samo eden (10 %) se je ocenil zadostno oziroma dobro; so pa štirje učenci, ki menijo, da dobro poznajo računalništvo v oblaku (40 %), povprečna ocena skupine

je 2,5. V naslednji oceni so učenci ocenjevali poznavanje virtualizacij in virtualizacijskih pristopov; kar štirje (40 %) učenci se še niso srečali s tem, toliko jih je tudi prepričanih v dobro poznavanje, eden (10 %) se je ocenil z oceno 2, drugi (19 %) pa s štiri, povprečna ocena je 2,2. Trije učenci (30 %) terminala še niso uporabljali, dva (20 %) poznata zgolj osnove, dva (20 %) ga srednje poznata, prav tako dva (20 %) zelo uporabljata terminal, samo eden (10 %) se je ocenil z odličnim poznavanjem. Polovica (50 %) vzorca uporablja virtualizacijska orodja, preostala polovica pa ne. Nobeden učenec še ni uporabljal platformo, ki omogoča izkoriščanje infrastrukture, prav tako nobeden še ni postavil kakšnega oblaka.

Glede na ocene prehodnih znanj in izkušenj predvidevamo, da je smiselno oblikovati dva enako velika vzorca. Tako bomo v nadaljevanju dobili primerne ocene tečaja z vidika izkušenih in neizkušenih učencev. V prvega bomo vključili učence s slabo podlago teoretičnega in praktičnega znanja, v nadaljevanju jo bomo imenovali skupina A; drugi vzorec pa je ravno obratno in ga bomo v nadaljevanju označili kot skupina B. Skupino A tvorijo učenci pod zaporednimi števkami 1, 3, 7, 8 in 9 iz Tabele 7.1, preostali pa so v skupini B. Omenjeni skupini bomo v nadaljevanju ovrednotili glede na povprečne vrednosti in ocene.

Kratek povzetek skupine A, ki si ga je mogoče pogledati na Sliki 7.1, pravi, da je povprečna starost učencev 22,2 leti, povprečna ocena skupine v poznavanju arhitekture je 2,2, v poznavanju informacijskih in telekomunikacijskih sistemov je 2, poznavanje oblaka je 1,6, povprečna ocena poznavanja virtualizacije je 1,4 in poznavanj uporabe terminala je 2. Povzetek skupine B si je prav tako mogoče pogledati na Sliki 7.1. Povprečna starost učencev je točno 26 let. Glede na status sta dva študenta, trije pa zaposleni kot računalničarji, povprečna ocena strinjanja s poznavanjem arhitekture računalniških sistemov je 4, poznavanja informacijskih sistemov je 3,4 in poznavanja oblaka je 3,8. Najslabša povprečna ocena v skupini je 3,2, s katero so ocenili poznavanje virtualizacije, poznavanje uporabe terminala v operacijskem sistemu Ubuntu

Linux so ocenili z oceno 3,8.



Slika 7.1: Značilnost obeh vzorcev glede na oceno predznanj in izkušenj.

## 7.3 Postopek zbiranja in obdelave podatkov

Za metodo zbiranja podatkov raziskovalnega vzorca smo uporabili dva vprašalnika, ki sta sistematično in načrtno zastavljena. V obeh so vključena enostavna in razumljiva vprašanja, ki so tako odprtega kot zaprtega tipa, in ocenjevalske lestvice, na podlagi katerih učenec oceni stopnjo strinjanja s trditvijo.

Pripravili smo dva vprašalnika, ki smo ju učencem zastavili pred in po izvedbi tečaja v učilnici Moodle. Obe skupini učencev sta odgovorili na ista vprašalnika. V vprašalniku pred izvedbo tečaja, ki je večinoma zaprtega tipa, smo dobili osnovne informacije o značilnosti vzorca, kot so spol, status in starost, ter podatke o predhodnem znanju s področja arhitekture računalniških sistemov, informacijsko-komunikacijskimi sistemi, računalništva v oblaku in virtualizacije ter s praktičnimi izkušnjami uporabe in namestitve katere izmed platform. Vprašalnik si je mogoče ogledati v Dodatku F. Iz vprašalnika, ki so

ga učenci izpolnili po opravljenem tečaju, smo dobili ocene pridobljenega znanja in razumevanja iz teoretičnega in praktičnega dela tečaja. V ta vprašalnik smo vključili še vprašanja odprtega tipa, s katerimi smo dobili povratne informacije o pravilni obliki tečaja, razumljivosti in ustreznosti navodil, zahtevnosti nalog, razumevanju namena posamezne naloge, primernosti tečaja za nalogo, kot je postavljanje oblaka, težavah učencev, njihovo mnenje o tečaju in predlogih za izboljšave. Vprašalnik si je mogoče ogledati v Dodatku F.

Z namenom preverjanja zastavljenih ciljev in ovrednotenja tečaja smo ocenili tudi podatke o rešeni nalogi, ki smo jih po tečaju zastavili učencem. Vključevala je podoben postopek, kot je bil prikazan v praktičnih učnih enotah. Tako smo preverili, če so učenci dosegli zastavljeni cilj.

Podatke smo statistično obdelali in jih pretvorili v ustrezno obliko za opisovanje raziskave. Številске podatke smo obdelali v programu Excel, kjer smo izračunali njihove aritmetične sredine in tako dosegli lažjo primerljivost med enim in drugim vzorcem učencev ter jih predstavili v obliki grafov. Opisne podatke, kot so spol in število učencev, smo predstavili z odstotki. Odgovore smo obdelali na osnovi ključnih besed in celotnega pomena odgovora.

## 7.4 Interpretacija rezultatov

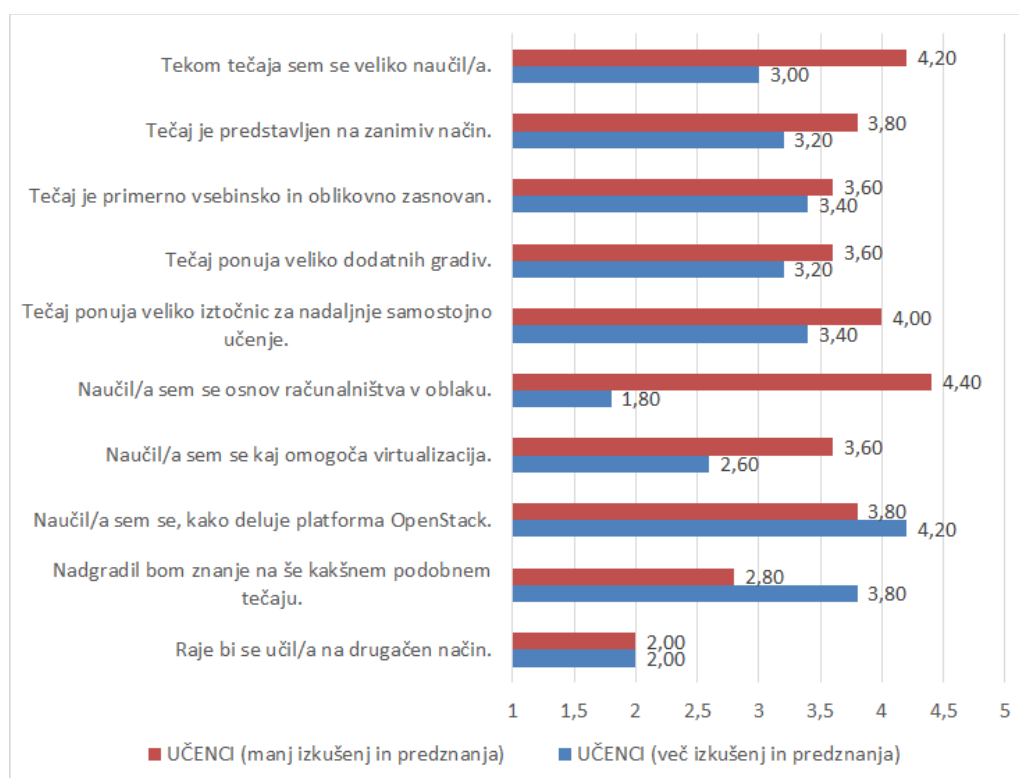
Vsak poučevalni način ima svoje slabosti in prednosti, bodisi učitelja pred tablo v predavalnici, samostojnega učenja ali pa e-učenja. V raziskavi smo ugotavljali:

- primernost in ustreznost e-učenja na primeru izdelanega tečaja;
- ustreznost in razumljivost praktičnih učnih enot;
- dosežnost ciljev.

V okviru evalvacije tečaja smo učencem omogočili oddaljeni dostop do že postavljene platforme OpenStack, preko spletnega vmesnika Horizon, ki ga

uporablja Laboratorij za računalniške komunikacije za raziskovalne in učne namene. Navodila za pripravo izvajalnega okolja so imeli učenci v spletni učilnici Moodle.

Primernost in ustreznost e-učenja smo ocenili s pomočjo podatkov, pridobljenih iz lestvice strinjanja, v kateri so učenci ocenjevali stopnjo strinjanja glede na trditve o izdelanem tečaju. Učenci so oceno podali glede na pet-stopenjsko lestvico, pri čemer ocena ena predstavlja popolno nestrinjanje s trditvijo, ocena pet pa, da se zelo strinjajo, kot je prikazano na Sliki 7.2.



Slika 7.2: Ocene stopenj strinjanj s trditvami o izvedbi e-tečaja.

Iz grafa je moč takoj določiti, da so mnenja obeh vzorcev zelo različna. Do največjega razhajanja pride pri trditvi koliko so se učenci naučili o računalništvu v oblaku. Skupina A (manj izkušenj in predznanja) meni, da se je naučila veliko, povprečna ocena je bila 4,4, medtem ko skupina B (več izkušenj in

predznanja) meni nasprotno s povprečno oceno 1,8. Ocene so podobne tudi pri ocenjevanju znanja iz virtualizacije; skupina A meni, da se je dobro naučila, podali so oceno 3,6, skupina B pa samo 2,6. Poznavanje delovanja platforme je skupina A ocenila z oceno 4,2, skupina B pa z oceno 3,8. Skupina A s 4,2 ocenjuje, da je pridobila veliko znanj, druga skupina pa je svojemu znanju dodala nekaj novega, oceno 3. Manj izkušenim učencem se zdi tečaj zanimiv, temu so podali oceno 3,8, drugi skupini je tudi zanimiv z oceno 3,2. Najmanjša razlika v ocenah je pri trditvi o vsebinski in oblikovni zasnovi; učenci z več izkušnjami ocenjujejo z oceno 3,4, učenci z manj pa z oceno 3,6. Učenci z manj predznanja se z oceno 3,6 strinjajo, da lahko svoje znanje nadgradijo v dodatnem gradivu, z oceno 3,2 pa druga skupina učencev. Bolj izkušeni učenci se z oceno 3,4 strinjajo, da tečaj ponuja veliko možnosti za nadaljnje delo, manj izkušeni pa z oceno 4 menijo, da lahko dobijo več znanja v dodatnih gradivih. Samo pri trditvi o drugačnem načinu učenja sta obe skupini enotni z oceno 2.

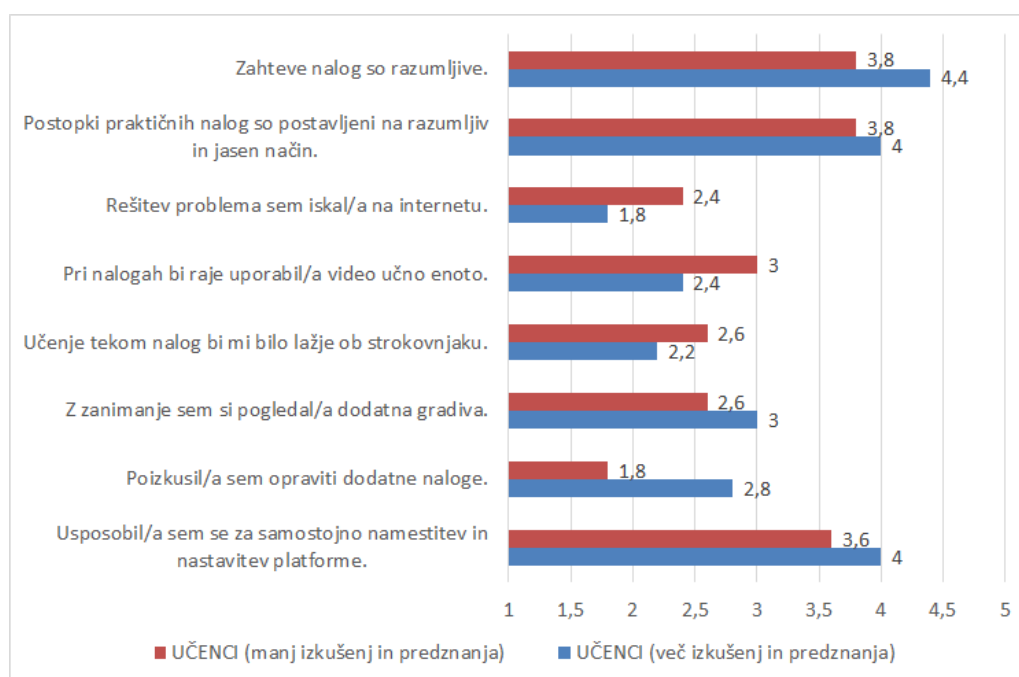
Na splošno je boljše ocene podal vzorec učencev, ki so se prvič srečali z računalništvom v oblaku in s postavljanjem platforme. Zanimivost, vsebina in oblika tečaja so zadovoljive iz obeh skupin, temu priča tudi dejstvo, da ni potrebno bistveno posegati v tečaj, saj je način učenja primeren. Skupina B je bila pri ocenjevanju veliko bolj skopa, kar je zaradi njihovih dosedanjih kompetenc razumljivo, saj v teoretičnem delu tečaja svojega znanja niso razširili. Tečaj ocenjujejo na splošno dobro. Po drugi strani je mlajša skupina teoretičnemu delu tečaja podala zelo visoko oceno. Obe skupini ocenjujeta, da razumeta, kako platforma deluje. Povprečna skupna ocena skupine A je 3,67, skupine B pa nekoliko manj 3,18.

V odgovorih o pomanjkljivostih tečaja je tudi nekaj nasprotij. Odziv na vprašanje o primerjavi tečaja s podobnimi tečaji, katerih so se učenci v preteklosti udeležili, so enotni, da je tečaj primerljiv s tečaji s podobno vsebino oziroma *»ni bistveno boljši«*. Eden izmed učencev je bil še posebej zadovoljen. Napisal je: *»Pri avtorju programa je možno zaslediti, da se je*



*poskušal vživeti v vlogo tečajnika, kar je vidno v natančnem popisu tečaja in nekaterih dodatnih vsebinah.*» Na vprašanje po spremembah v tečaju jih je nekaj, natančneje štirje, pripomnilo, da je tečaj dobro zasnovan, vendar bi raje opravljali tečaj, kjer bi bile vsebine zajete v video učne enote. Eden izmed njih je napisal: »Zanimiva se mi zdi ideja o kombinaciji učenja z video in pisnimi vsebinami (zgled videolectures). Primer: video – povzetek snovi in koraki instalacije, pisni del - povzetek ukazov in pomembnejših točk poglavja«.

V nadaljevanju si bomo pogledali rezultate raziskave na Sliki 7.3 o praktičnem delu tečaja, v katerem so se učenci poučili o namestitvi in nastavitvah platforme.



Slika 7.3: Ocene stopenj strinjanj s trditvami o praktičnem delu tečaja.

V grafu je moč opaziti, da so ocene strinjanj bolj skladne kot v prejšnjem grafu. Skupina A razumljivost zahtev nalog ocenjuje s 3,8, izkušena skupina pa s 4,4. Prav tako se obe skupini strinjata, da je postopek nalog razumljiv in jasen, manj izkušeni z oceno 3,8, izkušeni z oceno 4. Pri reševanju nalog

so si več pomagali manj izkušeni učenci, ki so strinjanje s tem ocenili 2,4, kot pa skupina B z oceno 1,8. Skupina A bi si za poučevanje namestitev in nastavitev platforme raje ogledala video učne enote, kot pa branje navodil, ocena strinjanja je 3; skupina B je ocenila strinjanje z 2,4. Podobno je tudi pri trditvi o prisotnosti strokovnjaka; skupina A je ocenila potrebo po strokovnjaku, ki bi jim pri nalogah svetoval, z oceno 2,6, skupina B z oceno 2,2. Vzorec z več predznanja, sodeč po oceni 3, si je bolje pogledal dodatna gradiva namestitev in nastavitev platforme, kot pa drugi vzorec, ki je skladnost s trditvijo ocenil z 2,6. Do največjih razlik med skupino A in skupino B pride pri opravljanju dodatnih nalog ter odločitvi o nadgradnji svojega znanja v prihodnje na podobnem tečaju. Opravljanje dodatnih nalog je skupina B ocenila z 2,8, medtem ko skupina A samo z 1,8. Oba vzorca menita, da sta se tekom opravljanja nalog usposobila za osnovno samostojno postavljanje platforme; ocena skupine A je bila manjša, in sicer 3,6, medtem ko je bila ocena skupine B 4.

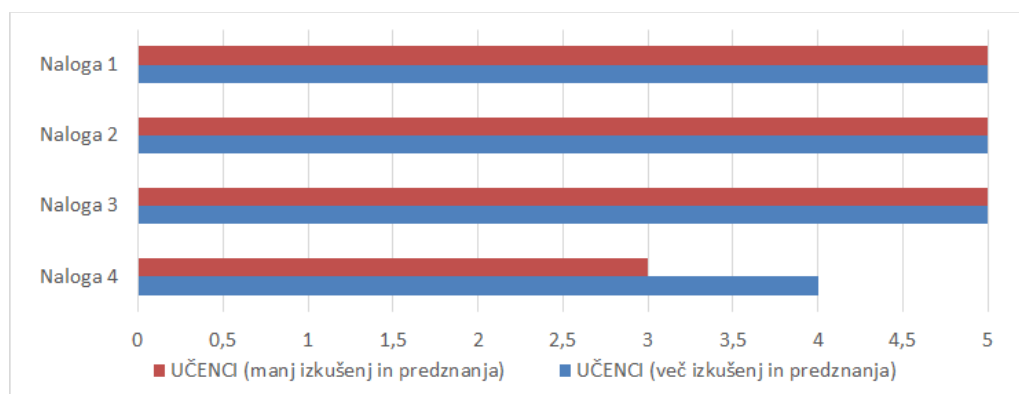
Obe skupini se strinjata, da so naloge v celoti dobro zasnovane, kar tudi nakazujejo ocene o usposobljenosti samostojne izvedbe tečaja. Nekateri so pri reševanju nalog posegli po rešitvah na spletu, večina učenci skupine A, ki bi prav tako želeli imeti postopke nalog v video učnih vsebinah. Nekaj jih tudi meni, da bi se lažje učili ob strokovnjaku. Izkušnost skupine B je mogoče zaznati, saj so si pogledali dodatna gradiva oziroma poskusili so opraviti dodatne naloge.

Odziv na vprašanja, povezana s spremembami v praktičnem delu tečaja, nam da vedeti, da je večina imela velike težave z izvajalnim okoljem. Veliko jih je imelo težave z nastavljanjem jezika na tipkovnicah, saj niso našli posebnih znakov, ki so se uporabljali v nastavitvah namestitvenih datotek, ker je privzeta tipkovnica angleška. Tako so veliko časa porabili za spremembo jezika tipkovnice; nekateri so med izvajanjem nalog celo nenehno spreminjali med slovensko in angleško tipkovnico. Eden je glede izvajalnega okolja napisal: *»Zaradi remote dostopa nastopita dve nevšečnosti. Ne dela kombinacija tipk*

*CTRL + W (išči v uredniku nano), saj s tem zapremo brskalnik (to je mogoče rešiti v firefoxu z addonom). Zelo zoprno je tudi, ker ni mogoče kopirati teksta v konzolo.*» Druga pomanjkljivost je ponovno uporaba videa, v katerem bi bile naloge jasno demonstrirane.

## 7.5 Preverjanje zastavljenih učnih ciljev

Želeli smo še preveriti, če učenci dosežejo zastavljeni cilj, to je samostojna postavitev platforme. Učencem smo podali nalogo, ki zahteva vsa znanja, pridobljena iz praktičnih učnih enot. Nalogo je možno prebrati v nadaljevanju. Učenci so svoje postavitve infrastrukture shranili v obliki posnetka navideznega stroja. Tako smo si zagotovili možnost preverjanja opravljenih nalog. Učenec je za uspešno izdelano nalogo prejel eno točko, za pol ali neizdelano naloge je dobil oceno 0. Na Sliki 7.4 si je moč ogledati uspešno opravljene naloge.



Slika 7.4: Rezultati pravilno opravljenih nalog.

Prvo, drugo in tretjo nalogo so vsi učenci opravili uspešno, v četrti nalogi sta dva učenca iz skupine A naredila nekaj napak, iz skupine B pa eden. Dva sta naredila napako v tretji nalogi pri namestitvah komponente Nova, zato v četrti nalogi nista mogla ustvariti omrežja za navidezne stroje in ju pognati. Eden od učencev se je zatipkal pri nastavitvah prijavnih datotek, zato ni

mogel izvajati ukazov omenjene komponente. Iz tega izhajamo, da bi lahko določenim delom v praktičnih enotah posvetili večji obseg obrazložitev.

Učencem smo zastavili spodnjo nalogo, na podlagi katere smo dobili rezultate pravilno opravljenih nalog.

*V podjetju Platforma d. o. o. (namišljeno podjetje) so se odločili spremeniti svoje poslovanje in opravila fizičnih strežnikov prenesti navidezne strežnike, ki jih omogoča platforma OpenStack. Pred kratkim so zaposlili skrbnika sistema IT, Miho, ki se prej še ni srečal s postavljanjem oblaka, in ga določili, da bo postavil omenjeno platformo. Da bi se »izvlekel« iz zagate, se je Miha odločil, da bo starejše kolege prosil, da mu naj naredijo potek namestitve v obliki seznama nalog. Glavni skrbnik podjetja Edi mu je pripravil naslednji seznam opravil:*

*1. naloga: pred začetkom nameščanja vsake platforme oziroma programske opreme je potrebno preverite minimalne strojne zahteve za njegovo stabilno delovanje, saj se v nasprotnem primeru kasneje lahko pojavijo velike težave. Glede na to, da še nimamo izkušenj z delovanjem platforme, jo namestite samo na en fizični strežnik. Če se tekom časa izkaže kot primerna rešitev za naše potrebe, ga bomo razširili še na preostale. V kolikor se še niste srečali z okoljem Ubuntu Linux, se seznanite z njegovo uporabo. Platforma pri svojem delovanju shranjuje različne podatke v podatkovno bazo in sporočilni strežnik, ki ju je potrebno ustrezno nastaviti. Zato priporočam prosto dostopno bazo MySQL in sporočilni strežnik RabbitMQ;*

*2. naloga: namestiti in nastaviti je potrebno posamezne komponente platforme. Za potrebe našega podjetja bood povsem dovolj komponente: Keystone, Glanc, Nova in Cinder. Vsako komponento je potrebno povezati s podatkovno bazo, pri nekaterih tudi s sporočilnim strežnikom, nastaviti avtentikacijske podatke in na koncu preveriti, če komponenta deluje. Ne pozabite nastaviti hipernadzornika KVM;*

*3. naloga: za operativno delovanje platforme morate ustvariti skrbnika sistema*

*in uporabnika ter za oba stanovalca in vlogo. Obema je potrebno nastaviti prijavne podatke. Potem je potrebno ustvariti storitve posameznih komponente, njihovega stanovalca ter dostopne točke API-jev posameznih storitev;*

*4. naloga: pred zagonom navideznega stroja je potrebno ustvariti navidezno omrežje, ki mu je potrebno določiti še način dostopa do navideznih strojev. Pred zagonom stroja je potrebno določiti sliko operacijskega sistema, uporabili bomo Ubuntu Linux, ki ga še prej prenesite s spleta, ter predlogo navidezne strojne opreme m1.small. Za dostop do stroja je potrebno ustvariti še SSH ključ. Z namenom preverjanja celotnega delovanja zaženite še navidezni stroj in preverite njegov status.*



## Poglavje 8

### Zaključek in nadaljnje delo

Glavni cilj diplomskega dela je bil uspešno postaviti platformo Openstack, ki je odprtokodni projekt za postavljanje oblaka. Omogoča nam ustvarjanje in upravljanje navideznih strojev oziroma računalnikov, katere je mogoče uporabljati tako kot navadne fizične računalnike. Glavna prednost uporabe platforme v poslovnem svetu se kaže v konsolidaciji obstoječega sistema IT in tako posledičnega prihranka.

V okviru diplomskega dela smo na podlagi psiholoških vidikov učnih procesov in priporočil priprave e-tečaja ugotovili, kako ga primerno zastaviti in oblikovati za samostojno učenje postavljanja infrastrukture, ki bi bil namenjen neizkušeni skupini uporabnikov, kot so bodoči skrbniki sistemov IT v ustanovah in podjetjih ter študentom računalništva. Zastavili smo tečaj, ki se v celoti izvaja v poučevalnem e-okolju Moodle na spletnem naslovu <http://rtk2014.fri.uni-lj.si/>. Razdelili smo ga na teoretični in praktični del. Tako smo dosegli sistematično in vodeno učno pot. Učne enote pretežno vsebujejo osnovno in dodatno učno gradivo ter iztočnice za samostojno učenje. V teoretičnem delu so učenci pridobili znanja iz računalništva v oblaku, virtualizacije in njenih tehnik ter teoretične vsebine platforme, v praktičnem pa sklop štirih nalog – namestitev in nastavitev platforme. Naloge so zasnovane tako, da udeležencem omogočajo hiter pregled potrebnih znanj in učnih ciljev,

na podlagi katerih se odločijo, če bodo opravljali nalogo; prav tako tečaj ponuja številne iztočnice za dodatne nastavitve platforme.

Z namenom ugotavljanja pomanjkljivosti tečaja smo opravili raziskavo na vzorcu desetih učencev. Pet učencev ni imelo dobrega predznanja, preostalih pet pa ravno nasprotno. Pred začetkom in po koncu tečaja smo jim zastavili vprašalnik. V prvem smo dobili značilnosti vzorca, kot so predznanje in izkušnje širše, povezane z računalništvom v oblaku, v drugem delu pa temeljito oceno tečaja, pomanjkljivosti in možnosti nadgradnje. Z namenom preverjanja učnih ciljev smo učencem zastavili praktični preizkus, ki zajema vse praktične učne enote, omogočili smo jim dostop do izvajalnega okolja. Naloge so oddali v obliki posnetkov slik navideznih strojev, v katerih smo pregledali namestitve in nastavitve ter pravilno delovanje platforme.

Kratek povzetek o rezultatih študije tečaja kaže primerno vsebinsko in oblikovno zasnovo z določenimi pomanjkljivostmi. Učenci so potrdili, da so navodila praktičnih nalog enostavna za razumeti, praktične učne enote potekajo v ustreznem zaporedju, postopki rešitev nalog so kvalitetni, naloge imajo jasno definiran cilj. Kot glavno pomanjkljivost tečaja so navedli nerodno okolje za izvajanje praktičnih učnih enot, saj je večina imela težave z vzpostavitvijo lokalizacije tipkovnice. Tako so s težavo poiskali določene posebne znake v nastavitvenih datotekah. V primeru izvedbe tečaja v vodenih video učnih vsebinah so učenci razdvojeni; nekateri menijo, da bi se več naučili preko videa, drugi pa so prepričani, da video ne bi odigral pomembne vloge, saj je v teoretičnem delu nepotreben in nima dodane vrednosti za prikazovanje dogajanja na terminalu; bolj je namenjen prikazovanju dogajanja na grafičnem uporabniškem vmesniku.

Zato smo po analizi in statistični obdelavi rezultatov ter njihovi interpretaciji opravili intervju z učenci, ki zagovarjajo izvedbo tečaja v video učnih enotah in so na podlagi tega primanjkljaja oddali najnižje ocene oziroma nasprotno skupino, ki je podala višje ocene o izvedbi in poteku tečaja. Takšne ocene so statistično ubežniki in zahtevajo dodatno obravnavo. Glede na rezultate



raziskave smo bili zelo prepričani, da bi veliko težav razrešili s preprostimi in sistematičnimi video vsebinami, kjer bi zajeli posamezne teoretične ter praktične učne enote, pripravo izvajalnega okolja ter po možnosti tudi dodatna gradiva. Presenečeno smo ugotovili, da temu ne bi bilo tako.

Dva učenca sta bila zelo kritična do video učnih vsebin, da so včasih časovno potratne, saj se jima »*ne da gledat*« celotnega videa. Pri tem navajajo, da pogosto iščejo samo »*nek podatek*«, torej specifičen del v video predavanju, ki bi pojasnil in razrešil njihov problem ter ga večinoma ne najdejo, zato bežno pregledajo vsebino. V kolikor ni mogoče poiskati samo tistega dela, v iskalniku spremenijo niz besed in ponovno iščejo video učne vsebine. Pri tem oba podobno poudarjata: »*Naslov zna biti zelo zavajajoč in je njegova vsebina povsem neka druga stvar.*« Eden še posebej ocenjuje, da polovica video vsebin na medmrežju za njihovo izmenjavo nima primerne uvoda in si najraje ogleda posnetek, ki traja do slabih štirih minut. Zelo pomembno mu je, če je v začetnih sekundah videa na voljo kratek spored obravnavanih vsebin. Zaradi omenjenih nevšečnosti se večkrat odločita poiskati podatke v tekstualnih učnih vsebinah, saj omogoča hitrejši celotni pregled vsebine – »*preskrolata*« stran.

Po drugi strani nekaterim učencem časovno daljši video posnetki ne predstavljajo velikega problema, »*z veseljem*« jih pogledajo oziroma vzporedno opravljajo podobno dogajanje, kot je prikazano v video vsebini. Pri tem jih najbolj moti, če se dogajanje na videu odvija prehitro in ne morejo usklajeno izvajati podobno opravilo. Še posebej jim je »*všeč*«, če avtor podaja razgibano tematiko, jo pojasnjuje s stališča praktičnih pogledov, celo izvede kakšen praktični preizkus in ga predvaja v zelo upočasnjem načinu, celostno primerja z ostalimi pristopi ali rešitvami in pozornost posveča malenkostim, ki niso vezane posredno na tematiko, kot so podnapisi, da se dogajanje »*za-zuma*« samo na določen del grafičnega vmesnika in se točno vidi, na katero ikono oziroma gumb je avtor kliknil. Za ta način pridobivanja znanj so se odločili zaradi tega, ker je bistveno manj naporen kot branje nekega učbenika

ali knjige. Pogosto so ga oziroma ga prakticirajo pri študiju. V en glas se strinjajo, da pri »*resnem delu*« kombinirajo video lekcije s študiranjem pomembnih delov v učbenikih oziroma knjigah.

Iz dveh omenjenih vzorcev je mogoče sklepati, da so tečaj opravljali učenci z različnimi predznanji, izkušnjami in s stopnjo samostojnosti, kar dokazuje tudi razpon učencev v letih, ki je med 22 in 29, vendar so svoje predznanje in izkušnje napačno ocenili. V primeru prvih dveh učencev je moč zaslediti, da imata veliko predznanja na področju računalništva in uporabe računalnika na splošno, vendar je v določenih situacijah njuno znanje nepopolno. Pri tem sta sposobna takojšnje identifikacije problema in iskanja njegove rešitve. Opazimo lahko tudi željo po instantnem efektu, torej takojšnji najdbi rešitve in njene uporabe, brez kakršnega koli časovno daljšega iskanja. Druga skupina učencev pa je nasprotno usmerjena – ni sposobna takojšnje prepoznave problema in natančnega iskanja rešitev. Namesto na čas se raje osredotoči na nova znanja, ki jih pridobijo v video predavanjih. Takšen način učenja jim predstavlja tudi zabavo.

Glede na rezultate dodatne raziskave bi bila rešitev v individualizaciji. Tako bi tečaja ustrezno prilagodili vsem zahtevam učencev. V sklopu le-tega bi pred začetkom izdelave tečaja opravili raziskavo, v kateri bi se osredotočili na širok spekter značilnosti učencev. V raziskavi bi jim kot izhodiščno točko podali naslov, smernice in zastavljene cilje tečaja, jih podrobno povprašali za primerno vsebino obeh delov tečaja, o predlogih o izvedbi posameznih delov tečaja, postopkih ter zastavljenosti in vsebini video lekcij. Tako bi tečaj sestavili v skladu z izkušnjami in na predhodnem znanju učencev. Zaradi tega bi bil bolj strukturiran po obsegu in vsebinah. Za naprednejše učence bi pripravili kratke povzetke učnih enot, za šibkejšje pa obsežne učna gradiva, ki bi jih vodila od osnov do zapletenih tematik in nastavitev.

Nadaljnja uporaba tečaja bi bila z nekaterimi modifikacijami primerna kot samostojen predmet ali pa v sklopu tematsko podobnega predmeta na naši fakulteti, saj imamo ustrezno akademsko osebje, ki se ukvarja s tematikami

s področja računalništva v oblaku. Tako bi lahko študente seznanili z eno izmed najbolj zanimivih smeri v razvoju informacijskih tehnologij v tem trenutku. Vključili bi lahko nadaljevalni sklop nalog, v katerem bi se učenci učili o dodatnih nastavitvah, orodjih ter integracijo OpenStack z drugimi platformami.



# Dodatek A

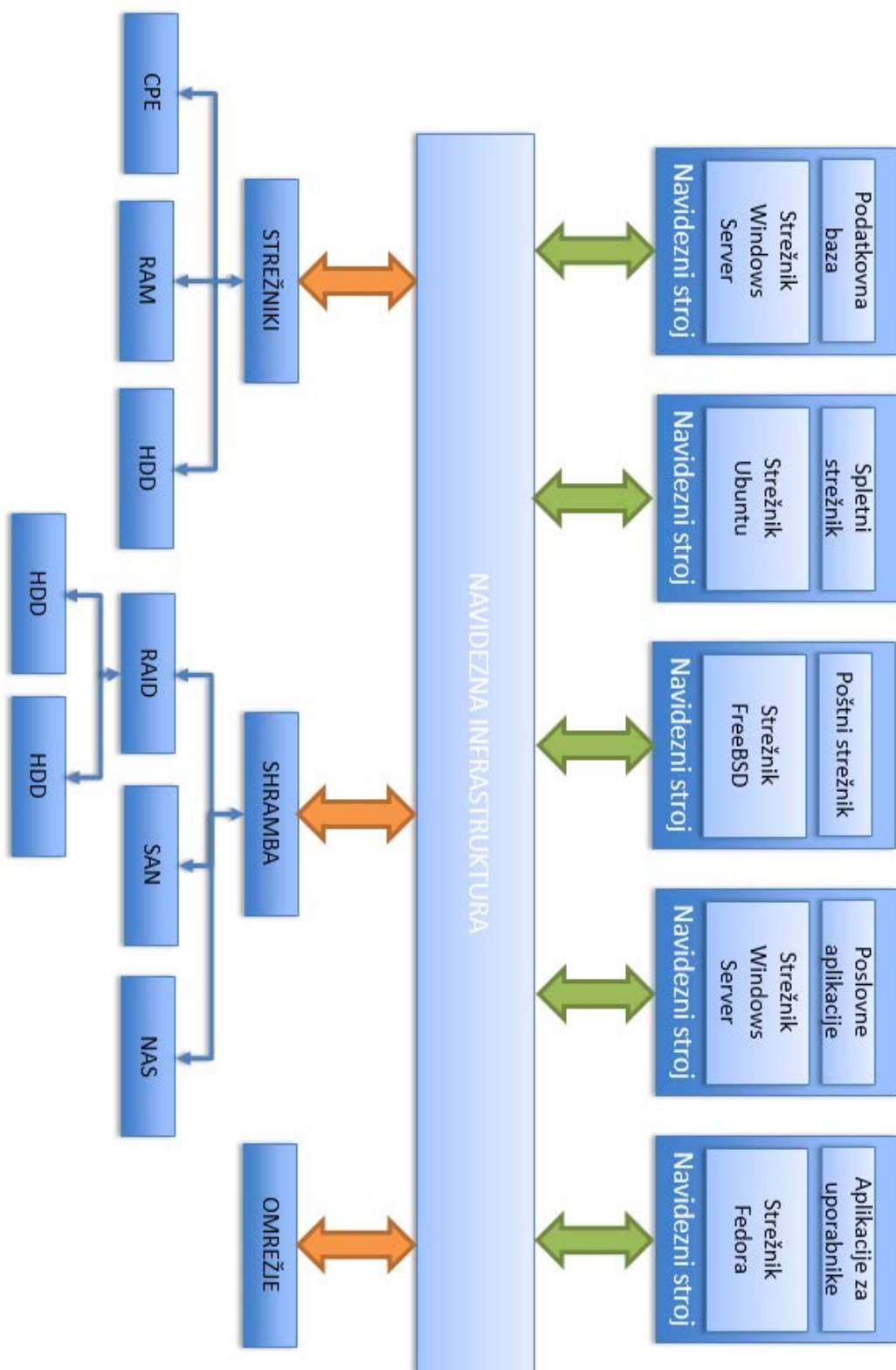
## Zgoščenska

Na priloženi zgoščenci je celoten tečaj v obliki varnostne kopije spletne učilnice Moodle.



## Dodatek B

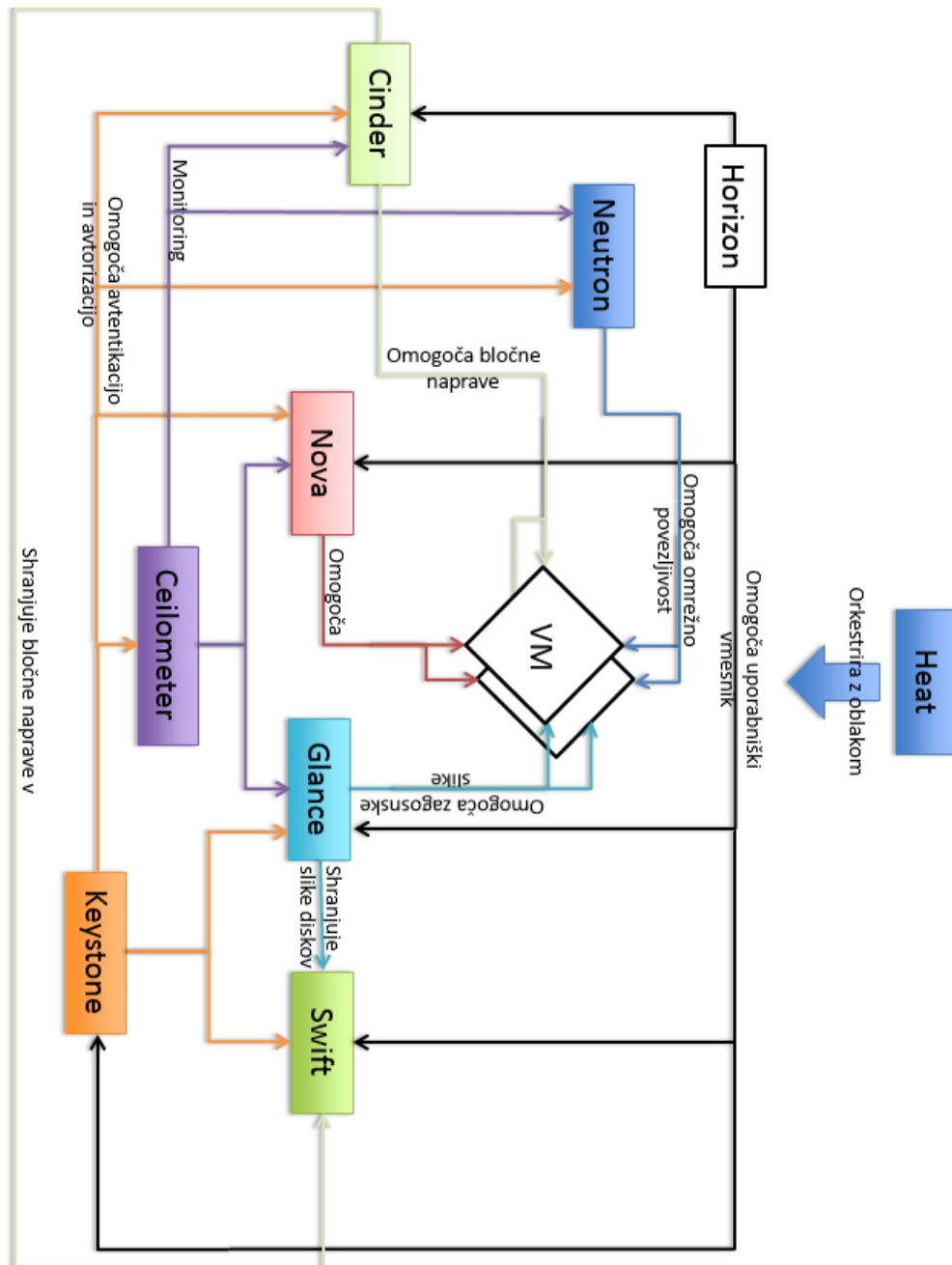
### Primer navidezne infrastrukture





## Dodatek C

### Konceptualna arhitektura platforme OpenStack



## Dodatek D

### Procesi komponent

V tem dodatku so procesi posameznih komponent platforme OpenStack.

Procesi komponente Nova:

PROCES	OPIS
nova-api-service	Sprejema zahteve končnih uporabnikov preko klicev API-ja. Upravlja z navideznimi stroji, omrežjem in slikami. Zato storitev Nova podpira tudi druge API-je kot so Googlov Compute API in Amazonov EC2 API.
nova-api-metadata service	Sprejema metapodatke navideznih strojev preko lastnega ali pa EC2 API-ja. Metapodatki so naslov IP, ID, tip, ime gostitelja in druge. API se uporablja v načinu z nameščeno storitvijo <i>nova-network</i> .
nova-compute-service	Prikriti proces sprejema operacije nad navideznim strojem iz sporočilne vrste in izvaja sistemske ukaze (primer zagon KVM instance). Podpira širok nabor hipernadzornikov.
nova-schedule	Sprejema zahteve iz vrste in glede na trenutno porazdelitev obremenitev na računskih strežnikih določi na katerem se bo navidezni stroj izvajal.
nova-object store	Prikriti proces, ki omogoča komunikacijo med Amazonovim vmesnikom S3 in vmesnikom za slikovno hrambo.
euca2ools client	Proces, ki prevaja iz ukazne vrstice Amazonovega EC2 API-ja v ukaze, ki se lahko izvajajo na platformi.

PROCES	OPIS
nova-conductor module	Vmesnik med procesom <i>nova-compute</i> in podatkovno bazo, tako omenjeni storitvi preprečuje neposreden dostop do baze. Namesti se ga na računski strežnik.
nova-cert daemon	Prikriti proces, ki služi za upravljanje s certifikati x509.
nova-network	Prikriti proces, ki je podoben procesu <i>nova-compute</i> in upravlja omrežje (nastavlja mostove <i>ang. bridge</i> , vzdržuje tabele IP).
nova-dhcpbridge	Shranjuje naslavljanje naslovov IP, ki se dodeljujejo navideznim strojem, v podatkovno bazo, tako vedno zagotovi enak naslov istemu navideznemu stroju.
nova-consoleauth daemon	Prikriti proces, ki avtenticira žetone uporabnikov. Žetone uporabljajo naslednja dva procesa.
nova-novncproxy daemon	Prikriti proces, ki omogoča povezavo VNC preko posrednega strežnika ( <i>ang. proxy</i> ) za dostop do izvajajočih navidezni strojev.
nova-xvpnvncproxy	Posredni prikriti proces, ki dostopa do izvajajočih navideznih strojev.
nova client	Ukazna vrstica za končne uporabnike oziroma skrbnike sistemov.
nova-manage client	Omogoča skrbnikom sistema veliko pravic in avtorizacije pri dostopu do oblaka.

Načini avtentikacije in avtorizacije komponente Keystone:

NAČINI AVTENTIKACIJE IN AVTORIZACIJE	OPIS
Uporabnik ( <i>ang. user</i> )	Digitalna prezentacija osebe, sistema ali storitve, ki uporablja storitev oblaka. S pomočjo avtentikacije se prepoznajo uporabniki in njihove nadaljnje zahteve. Skrbnik sistema dodeli dostopne pravice uporabniku, ki določajo katere operacije lahko izvaja in količino virov, ki jih ima na voljo. V kolikor uporabnik želi dostopati do virov, mora biti dodan stanovalcu.
Prijavni podatki ( <i>ang. credentials</i> )	Poverilnice so podatki, ki pripadajo točno določenemu uporabniku s katerimi se avtentificirajo in ne smejo biti dostopni tretjim osebam. Primer: uporabniško ime in geslo, ujemanje uporabniškega imena in ključa APIja ter uporabniški žetoni.
Avtentikacija ( <i>ang. authentication</i> )	Potrjevanje identitete uporabnika in njegovih zahtev. V kolikor se potrdi identiteta uporabnika, potem se njegove zahteve sprejeme in izvede. Primer podatkov so prijavni podatki.
Žeton za dostop ( <i>ang. token</i> )	Žeton vsebuje zapis o varnostnih elementih storitve, procesa in dostopih do določenih virov. Žeton po določenem času poteče.

NAČINI AVTENTIKACIJE IN AVTORIZACIJE	OPIS
Stanovalec ( <i>ang. tenant</i> )	Stanovalec določa količino virov za uporabnike, račune, organizacije in projekte. Viri so VLAN, fiksni in plavajoči naslovi IP, število navideznih naprav, število navideznih strojev, število vCPE, slike navideznih strojev, kapacitete navideznih naprav in uporabnike.
Storitev ( <i>ang. service</i> )	Storitvam platforme je potrebno dolčiti končne točke preko katerih uporabnik lahko dostopa do posameznih komponent in jih uporablja.
Dostopna točka ( <i>ang. endpoint</i> )	Dostopna točka ali končna točka v omrežju je računalnik ali konzola v omrežju, ki ima dostop do storitev oblaka. Določa jo naslov storitev URL.
Vloga ( <i>ang. role</i> )	Skrbnik sistema dodeli vlogo uporabniku in tako določi kateri uporabnik ima dostopne pravice do podatkov, storitev in katere operacije lahko izvaja.

Procesi komponente Glance:

PROCES	OPIS
glance-api	Sprejema ukaze API-ja za odkrivanje, proizvodovanje in shranjevanje slik diskov navidezni strojev.
glance-registry	Shranjuje, proizvaja in procesira meta podatke o slikah (velikost slike, tip slike, ID slike in druge).

Procesi komponente Neutron:

PROCES	OPIS
neutron-server	Sprejema zahteve in jih pošilja vtičem ( <i>ang. plug in</i> ) znotraj omrežja.
omrežni vtičniki in agenti	Omogočajo in onemogočajo vrata, ustvarjajo omrežja in podomrežja, izvajajo IP naslavljanje. Agenti so L3 ( <i>ang. Layer 3</i> ), DHCP in drugi agenti.
sporočilne vrste	Prenašajo sporočila med strežnikom <i>neutron</i> in agenti. Shranjujejo stanje omrežij v podatkovno bazo.



Procesi komponente Cinder:

PROCES	OPIS
cinder-api	Sprejema zahteve API-jev in jih posreduje storitvi <i>cinder-volume</i> za nadaljnje procesiranje.
cinder-volume	Shranjuje in bere stanje navideznih naprav v/iz baze. Sodeluje z ostalimi procesi ( <i>cinder-scheduler</i> ) preko sporočilne vrste.
cinder-scheduler	Prikriti proces, ki določi strežnik, na katerem je optimalno prostora za tvorjenje novih navideznih naprav. Prav tako uporablja sporočilne vrste z namenom posredovanja sporočil med procesi Cinderja, shranjuje stanja navideznih naprav v bazo in Novi omogoča navidezne naprave za instance.
sporočilne vrste	V sporočilno vrsto se postavijo sporočila, ki jih obdelujejo storitve Cinderja.

Procesi komponente Swift:

PROCES	OPIS
swift-proxy-server (account server)	Posredniški strežnik za račune, ki so definirani za shranjevanje objektov. Sprejema ukaze swiftovega API-ja in zahteve HTTP za nalaganje slik, spreminjanje metapodatkov in izdelovanje vsbovalnikov.
swift-account-server (container server)	Upravlja z računi, ki jih definira Swift.
swift-container-server (object server)	Strežnik, ki skrbi za upravljanje in mapiranje vsebovalnikov ter objektov.
swift-object-server	Upravlja z dejanskimi objekti, kot so datoteke na shranjevalnih vozliščih oziroma strežnikih ( <i>ang. storage node</i> ).
periodični procesi	Skrbijo za odstranjevanje podatkov na velikih podatkovnih shrambah ( <i>ang. data store</i> ). Na ta način zagotavlja konsistentnost in razpoložljivost gruče strežnikov. Ostali periodični procesi izvajajo revizije, posodobitve in ponavljanje opravila, dokler opravila niso zaključena.
vmesnik WSGI	Opravlja avtentikacijo za dostop do Swifta, ki je ponavadi Keystone.

Procesi komponente Ceilometer:

PROCES	OPIS
ceilometer-api	Zbira podatki o statistiki uporabe vseh računskih strežnikov za naplačevalne sisteme in orodja za analizo podatkov.
ceilometer-agent-central	Zbira statistiko uporabe na nadzornem vozlišču (ni povezano z navideznimi stroji in računskimi vozlišči).
ceilometer-agent-compute	Od hipernadzornika pridobiva podatke o njegovem delovanju.
ceilometer-agent-notification	Sprejema sporočila AMQP od ostalih komponent platforme.
ceilometer-collector	Zbira sporočila AMQP od agentov in jih dostavlja procesu za beleženje podatkov.
ceilometer-alarm-notifier	Zbira statistiko z več nadzornih vozlišč in omogoča nastavitve sporočil o prekomerni uporabi storitev, doseženih pragovih prenosa podatkov in uporabe navideznih strojev.
podatkovna hramba	Upravlja in obdeluje več sočasnih branji in pisanji, ki jih prejema od storitve za telemetrijo.

Procesi komponente Heat:

PROCES	OPIS
heat CLI	Komunicira s procesom <i>heat-api</i> in omogoča zaganjanje AWS CloudFormation API-jev, ki so predloga infrastrukture.
heat-api component	Proces REST API, ki sprejema zahteve namenskega OpenStackovega REST API-ja in jih pošilja heat-engine preko zahtev RPC ( <i>ang. remote procedure call</i> ).
heat-api-cfn component	Proces je AWS Query API, ki je združljiv z AWS CloudFormation in zahtevo sprecesira do heat-engine preko klicev RPC.
heat-engine	Zaganja infrastrukturo iz predlog in se odziva na klice klijentevega API-ja.

Procesi komponente Trove:

PROCES	OPIS
python-troveclient	CLI, ki komunicira z <i>trove-api</i> .
trove-api	Omogoča RESTful API preko katerega upravlja z navideznimi stroji.
trove-conductor	Izvaja se na strani gostitelja in sprejema podatke od navideznih strojev.
trove-taskmanager	Upravlja s sistemom, ki izvaja operacije nad navideznimi stroji.
trove-guestagent	Nahaja se na gostujočem navideznem stroju in izvaja operacije nad podatkovno bazo

## Dodatek E

# Logična arhitektura platforme OpenStack



# Dodatek F

## Vprašalnika

Vzorec vprašalnika, ki smo ga zastavili pred izvajanjem tečaja:

### VPRAŠALNIK A

Vprašalnik je sestavljen iz 11-ih vprašanj oziroma trditev, na katere je potrebno odgovoriti z ocenami od 1 do 5; odgovor je potrebno izbrati glede na lastno mnenje ali pa ga napisati.

Ocena 1 predstavlja "NE POZNAM", 2 predstavlja "MALO POZNAM", 3 predstavlja "SREDNJE POZNAM", 4 predstavlja "DOBRO POZNAM" in 5 predstavlja "ZELO DOBRO POZNAM".

Preden se odločite, dobro premislite o svojih znanjih. V dodatno pomoč so vam dopisana pojasnila pod trditvami ali vprašanji.

**Vprašanje 1:** Spol:

a) ženski

b) moški

**Vprašanje 2:** Starost udeleženca tečaja: \_\_\_\_\_

**Vprašanje 3:** Status:

- a) študent
- b) zaposlen
- c) drugo

**Vprašanje 4:** Kako bi ocenili svoje znanje poznavanja računalniške arhitekture?

NE POZNAM    1    2    3    4    5    ZELO DOBRO POZNAM

**Vprašanje 5:** Kako bi ocenili svoje znanje poznavanja informacijsko - telekomunikacijskih sistemov?

NE POZNAM    1    2    3    4    5    ZELO DOBRO POZNAM

**Vprašanje 6:** Kako bi ocenili svoje znanje teorije računalništva v oblaku?

NE POZNAM    1    2    3    4    5    ZELO DOBRO POZNAM

**Vprašanje 7:** Kako bi ocenili svoje znanje povezano s virtualizacijo in virtualizacijskimi tehnikami?

NE POZNAM    1    2    3    4    5    ZELO DOBRO POZNAM

**Vprašanje 8:** Kako bi ocenili svoje znanje uporabe terminala v operacijskih sistemih Ubuntu Linux?

NE POZNAM    1    2    3    4    5    ZELO DOBRO POZNAM

**Vprašanje 9:** Ali v svojem okolju uporabljate orodja za virtualizacijo, kot so VirtualBox, VMware Player, VMWare Workstation ali katera druga?

- a) DA



b) NE

**Vprašanje 10:** Ali uporabljate katero platformo?

a) OpenStack

b) OpenNebula

c) VMware vCloud

d) Amazon AWS

e) Google Compute Engine

f) drugo \_\_\_\_\_

g) nobeno

**Vprašanje 11:** Ali ste katero platformo za ponujanje infrastrukture kot storitve že poskušali namestiti na računalnik oziroma strežnik?

a) OpenOstack

b) OpenNebula

c) CloudStack

d) Eucalyptus

e) drugo \_\_\_\_\_

f) nobeno

Vzorec vprašalnika, ki smo ga zastavili po izvajanju tečaja:

### VPRAŠALNIK B

Vprašalnik je sestavljen iz 22-ih vprašanj oziroma trditev, na katere je potrebno odgovoriti z ocenami od 1 do 5 oziroma napisati odgovor.

Ocena 1 predstavlja "SE NE STRINJAM", 2 predstavlja "MALO SE STRINJAM", 3 predstavlja "SREDNJE SE STRINJAM", 4 predstavlja "SE STRINJAM" in 5 predstavlja "ZELO SE STRINJAM".

Preden se odločite, dobro premislite o svojih znanjih. V dodatno pomoč so vam dopisana pojasnila pod trditvami ali vprašanji.

**Trditev 1:** Tekom tečaja sem se veliko naučil/a.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 2:** Tečaj je predstavljen na zanimiv način.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 3:** Tečaj je primerno vsebinsko in oblikovno zasnovan.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 4:** Tečaj ponuja veliko dodatnega gradiva.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 5:** Tečaj ponuja veliko iztočnic za nadaljnje samostojno učenje.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 6:** Raje bi se učil/a na drugačen način.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 7:** Naučil/a sem se osnov računalništva v oblaku.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 8:** Naučil/a sem se kaj omogoča virtualizacija.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 9:** Naučil/a sem se, kako deluje platforma OpenStack.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 10:** Usposobil/a sem se za samostojno namestitev in nastavitev platforme.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 11:** Zahteve nalog so razumljive.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 12:** Postopki praktičnih nalog so postavljeni na razumljiv in jasen način.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 13:** Rešitev problema sem iskal/a na internetu.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 14:** Pri učenju bi raje uporabil/a video lekcijo namesto branja in prepisovanja ukazov.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 15:** Učenje tekom nalog bi mi olajšal strokovnjak, ki bi spremljal opravljanje nalog, ustrezno svetoval in pomagal.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 16:** Z zanimanje sem si pogledal/a dodatna gradiva.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 17:** Poizkusil/a sem opraviti dodatne naloge.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Trditev 18:** V prihodnosti se bom udeležil/a še kakšnega tovrstnega tečaja in nadgradil/a svoje znanje.

SE NE STRINJAM    1    2    3    4    5    ZELO SE STRINJAM

**Vprašanje 19:** Kaj bi spremenili pri teoretičnem delu tečaja?

---

---

**Vprašanje 20:** Kaj bi spremenili v praktičnem delu tečaja?

---

---

**Vprašanje 21:** Kaj ste pogrešali pri tečaju?

---

---

**Vprašanje 22:** Kako bi primerjali ta tečaj z ostalimi tečaji, ki ste se jih udeležili, s spletnimi učilnicami in z video lekcijami?

---

---



# Literatura

- [1] “OpenStack Cloud Administrator Guide”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://docs.openstack.org/admin-guide-cloud/content/>.
- [2] “Amazon Web Service (AWS) – Cloud Computing Services”. Amazon Inc., Zadnji dostop, dne 18.10.2014, na: <http://aws.amazon.com/>.
- [3] “AMD Virtualization”. AMD Inc. , Zadnji dostop, dne 18.10.2014, na: <http://www.amd.com/en-us/solutions/servers/virtualization>.
- [4] “Venture Technologies (formerly ISC) –AaaS (Aplication as a Service)”. Venture Technologies Inc.,Zadnji dostop, dne 18.10.2014, na: <http://www.isccorp.net/aaas.html>.
- [5] “App Engine - Google Cloud Platform”. Google Inc., Zadnji dostop, dne 18.10.2014, na: <https://cloud.google.com/appengine/>.
- [6] “Welcome to The Apache Software Foundation!”. Apache Software Foundation, Zadnji dostop, dne 18.10.2014, na: <http://www.apache.org/>.
- [7] “Apache CloudStack: Open Source Cloud Computing”. The Apache Software Foundation, Zadnji dostop, dne 18.10.2014, na: <http://cloudstack.apache.org/>.
- [8] “System architecture”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: [http://docs.openstack.org/admin-guide-cloud/content/section\\_telemetry-system-architecture.html](http://docs.openstack.org/admin-guide-cloud/content/section_telemetry-system-architecture.html).

- 
- [9] “Chapter 1. Architecture - OpenStack Installation Guide for Ubuntu 12.4/14.04 LTS”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: [http://docs.openstack.org/icehouse/install-guide/install/apt/content/ch\\_overview.html](http://docs.openstack.org/icehouse/install-guide/install/apt/content/ch_overview.html).
- [10] “Azure: Microsoft’s Cloud Platform”. Microsoft Corporation, Zadnji dostop, dne 18.10.2014, na: <https://azure.microsoft.com/en-us/>.
- [11] A. Escalante B. Furht. *Cloud computing fundamentals*. Springer, 2010.
- [12] J. Bervar. Nil-ov vodnik po storitvah računalništva v oblaku, marec 2014. Zadnji dostop, dne 18.10.2014, na: <http://www.nil.si/assets3554/wp-content/uploads/2013/04/nil-ov-vodnik-po-oblaku.pdf>.
- [13] “Introduction to the Block Storage service”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: [http://docs.openstack.org/icehouse/config-reference/content/section\\_block-storage-overview.html](http://docs.openstack.org/icehouse/config-reference/content/section_block-storage-overview.html).
- [14] “IBM Archives: 7094 Data Processing System”. IBM Corporation, Zadnji dostop, dne 18.10.2014, na: [http://www.ibm.com/ibm/history/exhibits/mainframe/mainframe\\_PP7094.html](http://www.ibm.com/ibm/history/exhibits/mainframe/mainframe_PP7094.html).
- [15] B. Brenčič. Računalništvo v oblaku: Stanje v sloveniji in primerjava s tujino. Diplomsko delo, Ekonomska fakulteta, Univerza v Ljubljani, 2010. Zadnji dostop, dne 18.10.2014, na: [www.cek.ef.uni-lj.si/UPES/brencic590.pdf](http://www.cek.ef.uni-lj.si/UPES/brencic590.pdf).
- [16] “EMC CaaS”. EMC Corporation, Zadnji dostop, dne 18.10.2014, na: <http://www.emc.com/collateral/hardware/white-papers/h8924-caas-vnx-vmax-wp.pdf>.
- [17] “OpenStack at Cisco”. Cisco System Inc., Zadnji dostop, dne 18.10.2014, na: <http://www.cisco.com/web/solutions/openstack/index.html>.
- [18] A. Clarke. *Designing computer-based learning materials*. Gower Publishing, Ltd., 2001.



- 
- [19] A. Clarke. *E-learning Skills*. Palgrave Macmillan, 2008.
- [20] “Cloud Computing - Gartner IT Glossary.
- [21] “Forrester Research : Topic : Cloud Computing”. Forrester Corporation, Zadnji dostop, dne 18.10.2014, na: <http://www.forrester.com/Cloud-Computing>.
- [22] “OpenStack Compute”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://docs.openstack.org/admin-guide-cloud/content/compute-service.html>.
- [23] “Compute Engine - Google Cloud Platform”. Google Inc., Zadnji dostop, dne 18.10.2014, na: <https://cloud.google.com/compute/>.
- [24] “OpenStack Configuration Reference”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://docs.openstack.org/admin-guide-cloud/content>.
- [25] “OpenStack Dashboard”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://www.openstack.org/software/openstack-dashboard/>.
- [26] “Database service overview”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://docs.openstack.org/icehouse/install-guide/install/apt/content/trove-service.html>.
- [27] “Google Drive”. Google Inc., Zadnji dostop, dne 18.10.2014, na: <https://www.google.com/drive/index.html>.
- [28] “EC2 compatibility API - OpenStack Cloud Administrator Guide – current”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://docs.openstack.org/admin-guide-cloud/content/instance-mgmt-ec2compat.html>.
- [29] “AWS — Amazon Elastic Compute Cloud (EC2) – Scalable Cloud Hosting”. Amazon Inc., Zadnji dostop, dne 18.10.2014, na:

<http://aws.amazon.com/ec2/>.

- [30] “Eucalyptus — Open Source Private Cloud Software”. Eucalyptus Systems Inc., Zadnji dostop, dne 18.10.2014, na: <https://www.eucalyptus.com/>.
- [31] G. Gerzon. “Intel Virtualization Technology Processor Virtualization”. Intel Corporation, Zadnji dostop, dne 18.10.2014, na: <https://software.intel.com/sites/default/files/m/0/2/1/b/b/1024-Virtualization.pdf>.
- [32] “Gmail - Email from Google”. Google Inc., Zadnji dostop, dne 18.10.2014, na: [https://www.gmail.com/intl/en\\_in/mail/help/about.html](https://www.gmail.com/intl/en_in/mail/help/about.html).
- [33] “Company – Google”, 2014. Google Inc., Zadnji dostop, dne 18.10.2014, na: <https://www.google.com/about/company/>.
- [34] “HP Official Site — Laptop Computers, Desktops, Printers and others”. Hewlett-Packard Company, Zadnji dostop, dne 18.10.2014, na: [www.hp.com/](http://www.hp.com/).
- [35] “IBM - United States”. IBM Corporation, Zadnji dostop, dne 18.10.2014, na: <http://www.ibm.com/us/en/>.
- [36] “OpenStack Identity concepts”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://docs.openstack.org/admin-guide-cloud/content/keystone-concepts.html>.
- [37] “OpenStack Image Service”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://docs.openstack.org/admin-guide-cloud/content/image-service-overview.html>.
- [38] “IOL Home - INFROMS”. INFROMS, Zadnji dostop, dne 18.10.2014, na: <https://www.informs.org/>.
- [39] “OpenStack Installation Guide for Ubuntu 12.04/14.04 (LTS)”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na:

<http://docs.openstack.org/icehouse/install-guide/install/apt/content/>.

- [40] W. Hu B. Johnson J. Matthews, T. Deshane. *Running Xen: a hands-on guide to the art of virtualization*. Prentice Hall PTR, 2008.
- [41] P. Hoffman K. Scarfone, M. Souppaya. "guide to security for full virtualization technologies". 2010. Zadnji dostop, dne 18.10.2014, na: <http://csrc.nist.gov/publications/nistpubs/800-125/SP800-125-final.pdf>.
- [42] D. Kodek. *Arhitektura in organizacija računalniških sistemov*. Bi-Tim, 2008.
- [43] "Logical architecture". OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://docs.openstack.org/admin-guide-cloud/content/logical-architecture.html>.
- [44] B. Marentič-Požarnik. *Psihologija učenja in pouka*. DZS, 2000.
- [45] "Microsoft US — Devices and Services". Microsoft Corporation, Zadnji dostop, dne 18.10.2014, na: [www.microsoft.com/](http://www.microsoft.com/).
- [46] M. Miller. *Cloud computing: Web-based applications that change the way you work and collaborate online*. Que publishing, 2008.
- [47] "Moodle - Open-source learning platform — Moodle.org. ". Moodle, Zadnji dostop, dne 18.10.2014, na: <https://moodle.org>.
- [48] G. E. Moore. "Moore's Law". Zadnji dostop, dne 18.10.2014, na: <http://www.moorelaw.org/>.
- [49] "MySQL :: The world's most popular open source database".
- [50] "Nas-San Comparison". Zerowait Corporation, Zadnji dostop, dne 18.10.2014, na: <http://www.nas-san.com/differ.html>.
- [51] "NASA". Zadnji dostop, dne 18.10.2014, na: [www.nasa.gov/](http://www.nasa.gov/).

- 
- [52] “Pacnet Offers First Pan-Asia Network-as-a-Service Architecture”. Pacnet, Zadnji dostop, dne 18.10.2014, na: <http://www.pacnet.com/2013/11/pacnet-offers-first-pan-asia-network-as-a-service-architecture/>.
- [53] “OpenStack Networking”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://docs.openstack.org/admin-guide-cloud/content/networking-service-overview.html>.
- [54] “Basic architecture with OpenStack Networking (Neutron)”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: [http://docs.openstack.org/havana/install-guide/install/yum/content/ch\\_overview.html](http://docs.openstack.org/havana/install-guide/install/yum/content/ch_overview.html).
- [55] “Object Storage characteristics”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: [http://docs.openstack.org/admin-guide-cloud/content/objectstorage\\_characteristics.html](http://docs.openstack.org/admin-guide-cloud/content/objectstorage_characteristics.html).
- [56] J. Onisick. “The Stack Wars: OpenStack vs. CloudStack - Network Computing”, maj 2012. Zadnji dostop, dne 18.10.2014, na: <http://www.networkcomputing.com/cloud-infrastructure/the-stack-wars-openstack-vs-cloudstack/a/d-id/1233664>.
- [57] “OpenNebula — Flexible Enterprise Cloud Made Simple”. OpenNebula Project, Zadnji dostop, dne 18.10.2014, na: [www.opennebula.org/](http://www.opennebula.org/).
- [58] “Home OpenStack Open Source Cloud Computing Software”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: [www.openstack.org](http://www.openstack.org).
- [59] “OpenStack Object Storage”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://docs.openstack.org/admin-guide-cloud/content/object-storage-service.html>.
- [60] “OpenStack”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://openstackabout.blogspot.com/>.

- 
- [61] “Orchestration service overview”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://docs.openstack.org/icehouse/install-guide/install/apt/content/orchestration-service.html>.
- [62] T. Grance P. Mell. The NIST definition of cloud computing. 2011. Zadnji dostop, dne 18.10.2014, na: <http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf>.
- [63] “PHP: Hypertext Preprocessor”. The PHP Group, Zadnji dostop, dne 18.10.2014, na: <http://php.net/>.
- [64] “Welcome to Python.org”. Python Software Foundation, Zadnji dostop, dne 18.10.2014, na: <https://www.python.org/>.
- [65] “Apache Qpid”. The Apache Software Foundation, Zadnji dostop, dne 18.10.2014, na: <https://qpid.apache.org/>.
- [66] “Rackspace: We manage your cloud services. You run your business”. Rackspace Inc., Zadnji dostop, dne 18.10.2014, na: [www.rackspace.com/](http://www.rackspace.com/).
- [67] V. Rebolj. “nekatera vprašanj motivacije pri učenju na spletu”, 2003. Zadnji dostop, dne 18.10.2014, na: <http://v-rebolj.com/teksti/e-izobrazevanje/motivacija.pdf>.
- [68] V. Rebolj. “kognitivni vidiki e-izobraževanja”, 2007. Zadnji dostop, dne 18.10.2014, na: <http://v-rebolj.com/teksti/e-izobrazevanje/kognitivni-vidik-e-izobrazevanja.pdf>.
- [69] “Simple Storage Service (S3) – Online Cloud Storage for Data and Files”. Amazon Inc., Zadnji dostop, dne 18.10.2014, na: <http://aws.amazon.com/s3/>.
- [70] M. Smith. “HyperText Markup Language”, januar 2013. Zadnji dostop, dne 18.10.2014, na: <http://www.w3.org/html/>.
- [71] “SOAP Specifications”. W3C, Zadnji dostop, dne 18.10.2014, na: <http://www.w3.org/TR/soap>.

- 
- [72] “AT&T Synaptic Storage as a Service - AT&T Cloud Services”. AT&T Inc., Zadnji dostop, dne 18.10.2014, na: [https://www.synaptic.att.com/clouduser/html/productdetail/Storage\\_as\\_a\\_Service.htm](https://www.synaptic.att.com/clouduser/html/productdetail/Storage_as_a_Service.htm).
- [73] J. Strickland. “How Cloud Computing Works”, april 2008. Zadnji dostop, dne 18.10.2014, na: <http://computer.howstuffworks.com/cloud-computing/cloud-computing.htm>.
- [74] V. Travassos. “Virtualization Trends Trace Their Origins Back to the Mainframe”, avgust 2014. Zadnji dostop, dne 18.10.2014, na: [http://www.ibmssystemsmag.com/mainframe/administrator/Virtualization/history\\_virtualization/](http://www.ibmssystemsmag.com/mainframe/administrator/Virtualization/history_virtualization/).
- [75] A. Kos U. Sedlar, J. Bešter. Računalništvo v oblaku v telekomunikacijah in primer uporabe. 2011. Zadnji dostop, dne 18.10.2014, na: <http://www.ltfe.org/objave/racunalnistvo-v-oblaku-v-telekomunikacijah-in-primeri-uporabe/>.
- [76] “The leading OS for PC, tablet, phone and cloud — Ubuntu”. Canonical Ltd., Zadnji dostop, dne 18.10.2014, na: [www.ubuntu.com](http://www.ubuntu.com).
- [77] “UsingTheTerminal”. Canonical Ltd., Zadnji dostop, dne 18.10.2014, na: <https://help.ubuntu.com/community/UsingTheTerminal>.
- [78] “User Stories » OpenStack Open Source Cloud Computing Software”. OpenStack Foundation, Zadnji dostop, dne 18.10.2014, na: <http://www.openstack.org/user-stories/>.
- [79] “Wikispaces”. Wikispaces, Zadnji dostop, dne 18.10.2014, na: <https://www.wikispaces.com/>.
- [80] “Web Services Description Language (WSDL)”. W3C, Zadnji dostop, dne 18.10.2014, na: <http://www.w3.org/TR/wsdl>.

- [81] “VMware: Software and Hardware Techniques for x86 Virtualization”. VMware Inc., Zadnji dostop, dne 18.10.2014, na: [http://www.vmware.com/files/pdf/software\\_hardware\\_tech\\_x86\\_virt.pdf](http://www.vmware.com/files/pdf/software_hardware_tech_x86_virt.pdf).